

CupCake Order Solution



- Deltagere
 - Jordy Sabimana, cph-js336@cphbusiness.dk, jordysabimana, A
 - Said Kayed, cph-sk249@cphbusiness.dk, saidkayed, A
 - Sercan Atici, cph-sa268@cphbusiness.dk, sercanatici, A
 - Tobias Jan Jensen, ,TobiasJensen1, A
- Projektet og rapporten blev lavet 16-03-2018

Indledning

Vi har udviklet en CupCake WebShop, der benytter MySql(Database), java servlets og jsp som backend, og html, css bootstrap som frontend.

WebShoppen sælger cupcakes, men kun som pick-up fra den butik man bestiller fra, kunden kan oprette en bruger, og logge ind på hans personlige side. Der på siden kan kunden vælge hvordan, kunden vil sammensætte cupcaken, og derefter bestille den til omgående levering.

Baggrund

Kort om Virksomheden

Virksomheden(CupCake Company), er multi international pick-up to go virksomhed. CupCake Company har et forældet System, hvor kunden selv skal ind i butikken, og vente bag de lange køer, for at bestille og bygge hans lækre cupcake, CupCake Company vil løse problemet, med deres nye WebShop, hvor kunden kan bestille antallet af cupcakes, og være klar inden kunden ankommer til butikken.

Virksomhedens Krav

Kunden skal kunne oprette en bruger, og kan logge ind, som kan derefter vil blive viderestillet til hans personlige side. I Kundens personlige side skal kunden kunne bestille hans/hendes Cupcake top og bund, hvor det kan blive gemt, i kundens indkøbskurv, som kunden kan også se,ændre på det og købe varen. Inde på indkøbskurven skal kunden kunne se varens pris, varens top og bund og antallet.

Teknologi valg

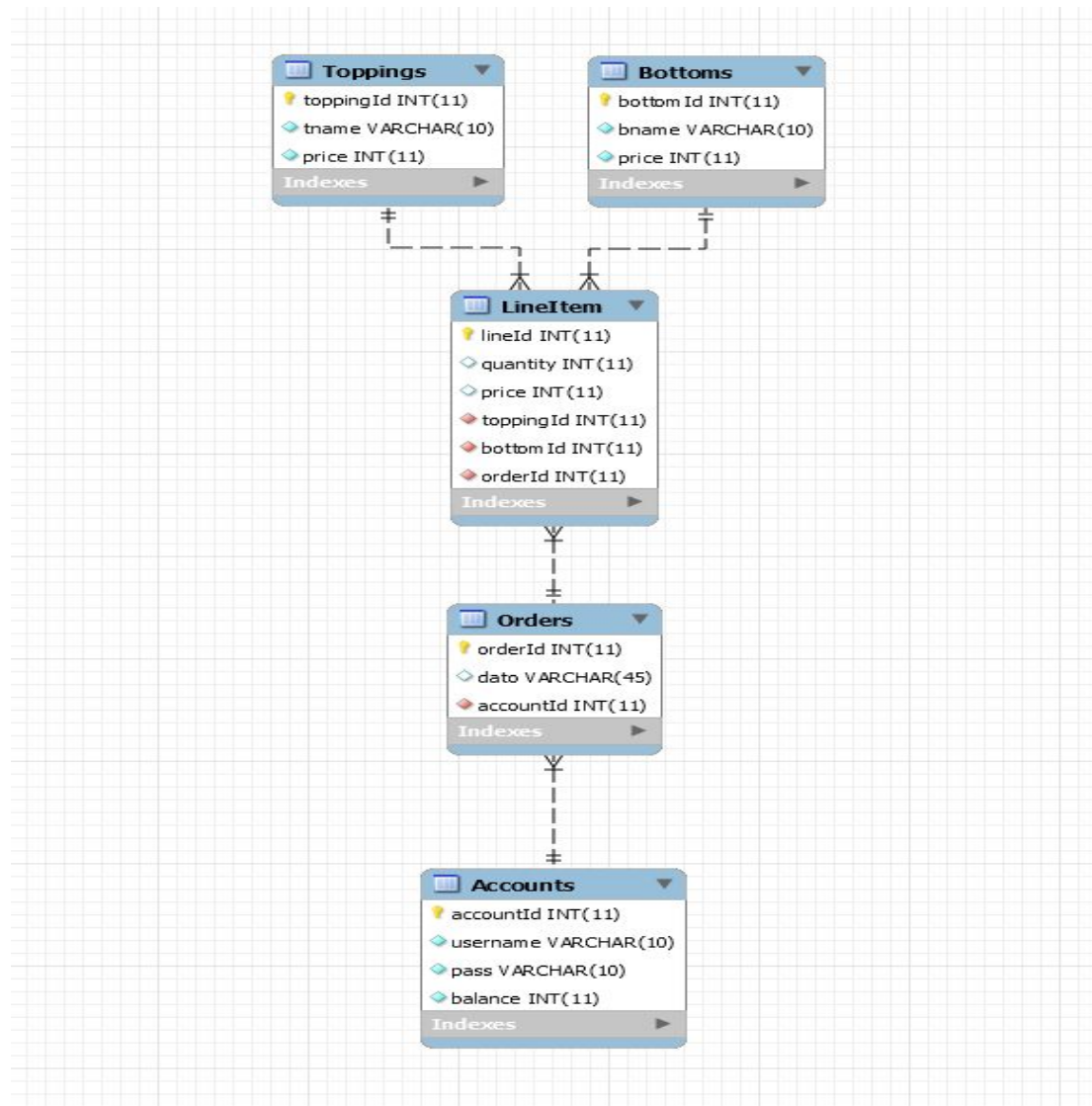
- Netbeans IDE 8.2
- MySQL server v14.14
- Ubuntu 16.04.3 - linux server
- Tomcat 8

Krav

- **Hvad er firmaets håb med dette system (hvad er deres vision for systemet eller hvilken værdi er det jeres system skal tilføre deres virksomhed).**

Virksomhedens mål er at få solgt så mange Cupcake som mulig og gøre det så nemt som mulig for kunden ved bestilling uden nogen form for forhindringer.

Domæne model og ER diagram



Man kan også supplere med en ShoppingCart klasse som ville have været i en domæne model.

ER diagrammet fortæller hvad forholdet mellem tabellerne er.

topping og bottoms har en til mange forhold med line items, man kan bestille en topping og en bottoms, men kan have mange i line items

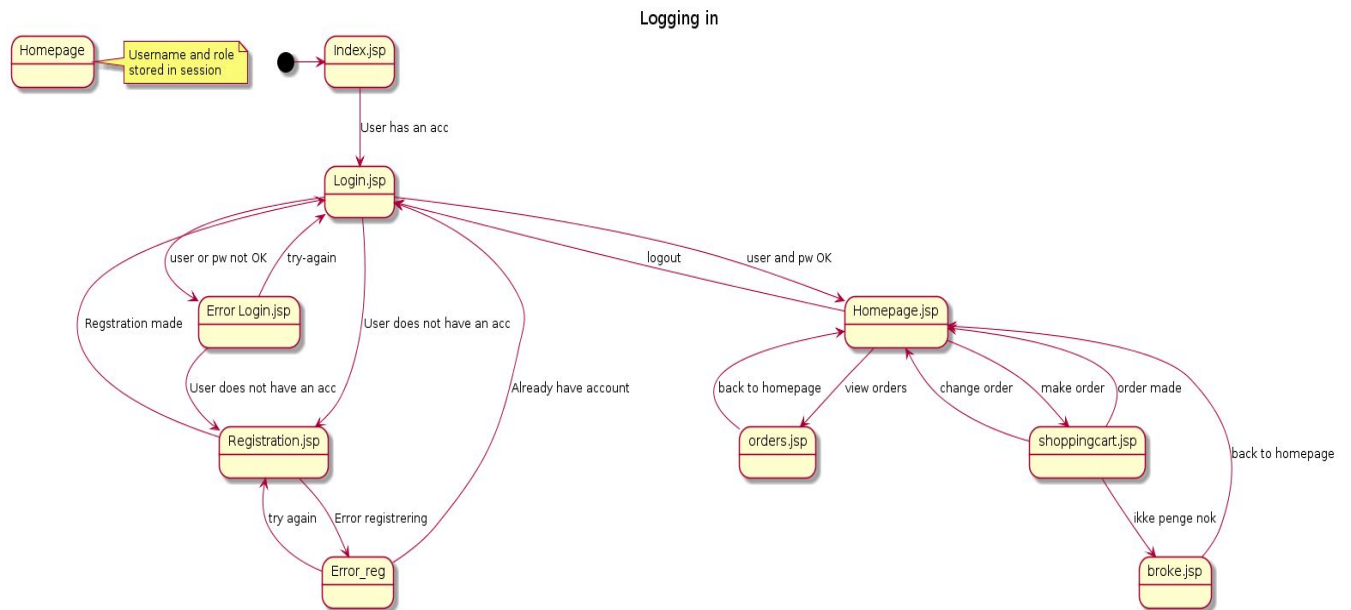
line items har mange til en forhold med orders

da line items bliver til en ordre.

orders har mange til en forhold med accounts,

da man kan have mange orders til en account, som man er logget ind med.

Navigationsdiagram



Navigationen fungerer som hjemmesidens skelet, og har til formål at gøre det nemt for brugeren at surfe rundt på hjemmesiden. Den skal kunne fjerne spørgsmål tegn og det skal være indlysende, hvor brugeren finder hvilke informationer.

jsp sider.

index.jsp, input.jsp, registration.jsp, homepage.jsp, shoppingcart.jsp, orders.jsp

Fejl sider(krav ikke opfyldes)

Broke.jsp, error_user_exists.jsp, regerror.jsp, wrongpass.jsp

index.jsp linker videre til homepage.jsp.

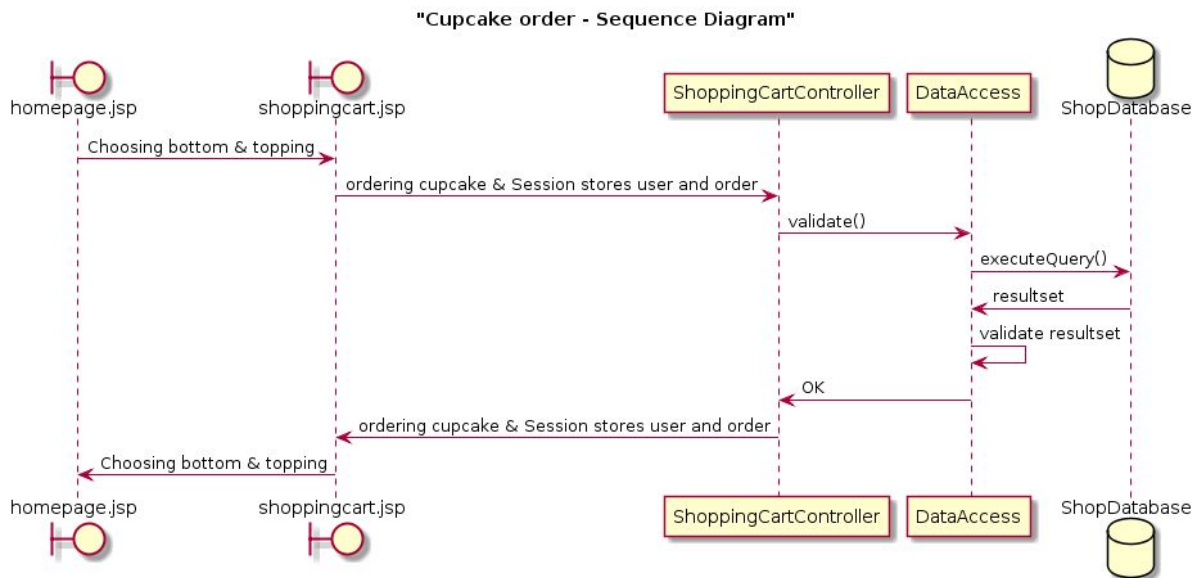
input.jsp sender videre til LoginController.

registration.jsp sender videre til RegistrationController.

shoppingcart.jsp sender videre til ShoppingCartController.

orders.jsp sender videre til OrderController.

Sekvens diagrammer



Sekvens diagrammet viser os fra hvor kunden er logged in og kan se kundeoplysningerne og kan vælge en top og bund for danne en cupcake.

Særlige forhold

- **Hvilke informationer gemmes i session**

User informationer
CupCake bestillinger

- **Hvordan man har valgt at lave validering af brugerinput.**

Vi benytter `doPost` metode som tjekker om username og password ikke er tomme og om username ikke allerede eksistere.

- **Hvordan man har valgt at lave sikkerhed i forbindelse med login.**

PreparedStatement for sikre sig imod sql injection.

- **Hvilke brugertyper, der er valgt i databasen, og hvordan de er brugt i jdbc**

Brugeren tob har kun adgang til cupbruger schema og kan oprette/redigere/slette indefor cupbruger schema.

Status på implementation

Dette afsnit skal liste hvor langt man er nået med implementationen. Typiske ting man kan have sprunget over er:

- Implementationen på ordre og at de ikke gemmes i sessionen
- Man kan ikke bestille flere af det samme produkt før man har købt det.
- Man kan ikke se sine gamle ordre.

Test

Der skal være lavet test. Vi har ikke i dette projekt haft fokus på at teste, så det skal ikke med. Du skal dokumentere tests ved at beskrive i tabel form:

Testen ligger i main klassen.

Klasser vi har testet

- Calculator
- DataAcces

Metoder vi har testet

- newAccount
- getUserByName
- getToppingName
- getBottomName
- getToppingsByld
- getBottomsByld
- updateBalance
- makeLineItem
- makeOrder