

iCupCake

Deltagere:

Jesper Rusbjerg

Pernille Halberg Lørup

Mette Rytte

Rasmus Helsgaun

E-Mail:

cph-jr211@cphbusiness.dk

cph-pl160@cphbusiness.dk

cph-mr383@cphbusiness.dk

cph-rh161@cphbusiness.dk

GitHub:

JesperRusbjerg

pernillelorup

metterytt

RasmusHelsgaun

2. semester - Datamatiker A-klasse

16. marts 2018



Indhold

1. Indledning.....	3
2. Baggrund.....	3
3. Teknologi	3
4. ER-diagram og domænemodel	4
Figur 4.1 Domænemodel:.....	4
Figur 4.2 ER-Diagram	5
5. Navigationsdiagrammer	6
Figur 5.1 Composite State Diagram for Login/Register:	6
Figur 5.2 Composite State diagram for kunde logget ind:.....	7
Figur 5.3 Composite State Diagram for admin logget ind:	8
6. Sekvensdiagram for cupcake-køb:.....	9
Figur 6.1 Sekvensdiagram.....	9
7. Særlige forhold	10
7.1 Session	10
7.2 Validering af brugerinput	10
7.3 Sikkerhed ved login.....	10
8. Status på implementation	10

1. Indledning

Dette projekt omhandler konstruktionen af en webshop, der sælger cupcakes.

Grundlæggende kan man som kunde i denne butik vælge mellem forskellige bottoms og toppings, som så bliver sat sammen til én cupcake. Man kan bestille flere cupcakes, og cupcakes i forskellige sammensætninger af bottoms og toppings. Det særlige ved denne butik er at éns ordre er klar til afhentning i samme øjeblik man har trykket 'Bestil' – altså ingen ventetid, men heller ingen levering.

2. Baggrund

Den (fiktive) virksomhed, som skal bruge dette system, har tidligere forsøgt sig med levering af cupcakes med posten – en løsning som af forskellige årsager viste sig ikke at fungere. Derfor skal nu denne webshop søsættes med en fornyelse af konceptet.

Krav til systemet omfatter:

- Man skal som kunde kunne registrere sig og siden logge ind for at handle i webshoppen
- Man skal kunne sammensætte cupcakes med valg af bottoms og toppings
- Betaling foregår ved at der sættes penge ind på kundens konto, noget som foreløbig bliver varetaget andetsteds. Dog skal dette system håndtere at trække det beløb fra kundens konto som der er handlet for
- Ligeledes skal systemet håndtere at man ikke kan handle for mere end det beløb der står på éns konto

3. Teknologi

Til projektet har vi anvendt følgende:

MySQL workbench 6.3 CE

MySQL Server version: 5.7.21-0 ubuntu0.16.04.1 (Ubuntu)

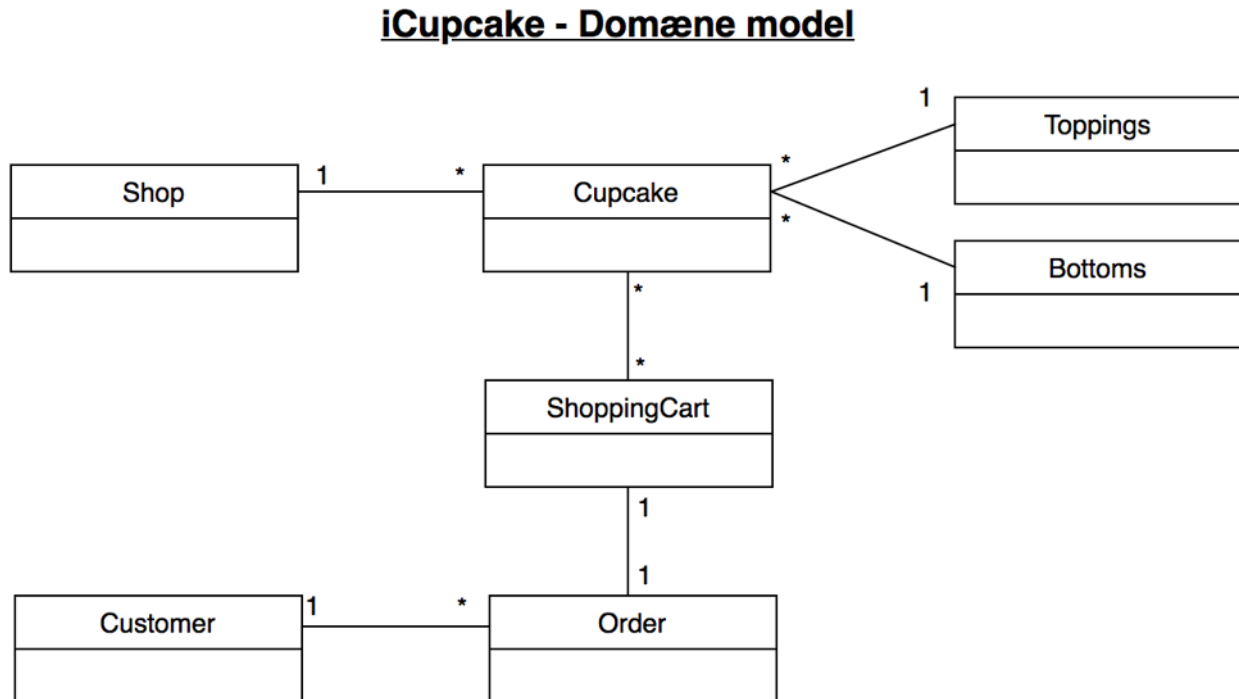
Netbeans 8.2 Patch 2

DigitalOcean Ubuntu 0.16.04.3

Apache Tomcat/8.0.32 (Ubuntu 0.16.04.2-b12)

4. ER-diagram og domænemodel

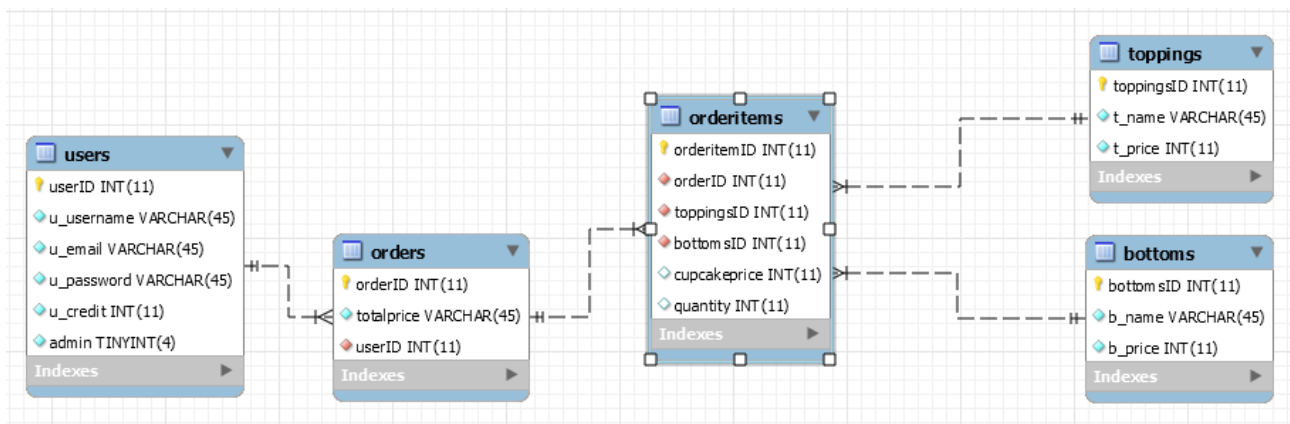
Figur 4.1 Domænemodel:



Domænemodellen for iCupCake viser her relationen mellem de omfattede objekter. Shoppen sælger mange forskellige cupcakes, som er sammensat af topping og bottom. Det er kun muligt at købe hele cupcakes, og altså ikke at købe separate toppings og bottoms. Hver cupcake har én topping og én bottom, mens hver henholdsvis topping og bottom kan høre til mange forskellige cupcakes. Hver cupcake kan indgå i mange shoppingcars, og en shoppingcart kan have mange cupcakes. Én shoppingcart indgår i én ordre, og endelig kan én kunde have mange ordrer.

En fyldestgørende beskrivelse af domænet fås ved at sammenholde domænemodellen med nedenstående ER-diagram:

Figur 4.2 ER-Diagram:

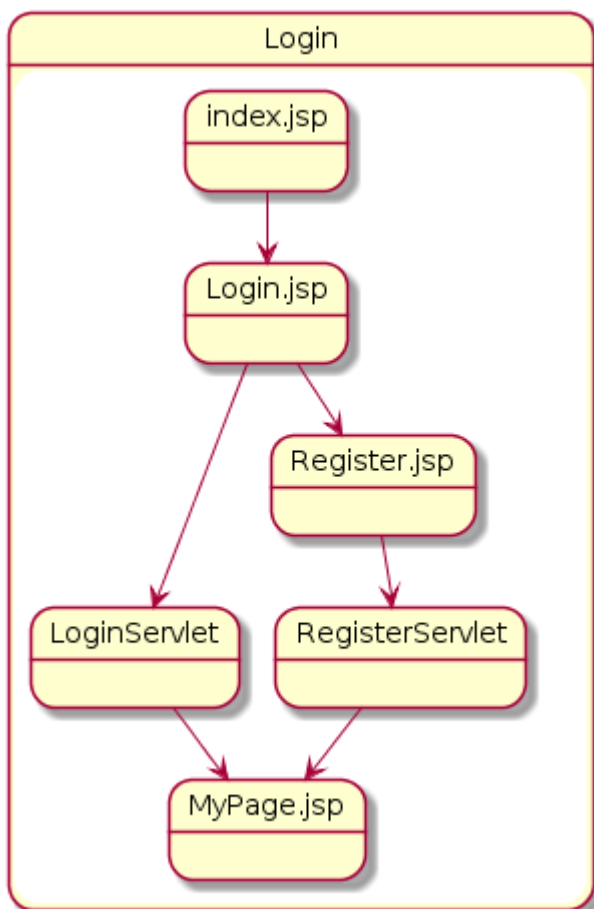


Alle tabeller i databasen benytter et autoincrementing ID som primærnøgle. De centrale entiteter i systemet udgøres af User, Order og Cupcake. I databasen er disse repræsenteret ved User i sin egen users-tabel, Order i sin egen orders-tabel og Cupcake er repræsenteret ved en opdeling i to tabeller, nemlig toppings og bottoms tabellerne. Endelig har vi orderitems-tabellen, der optræder som mellemlid i den mange-til-mange relation, der eksisterer mellem Order og Cupcake.

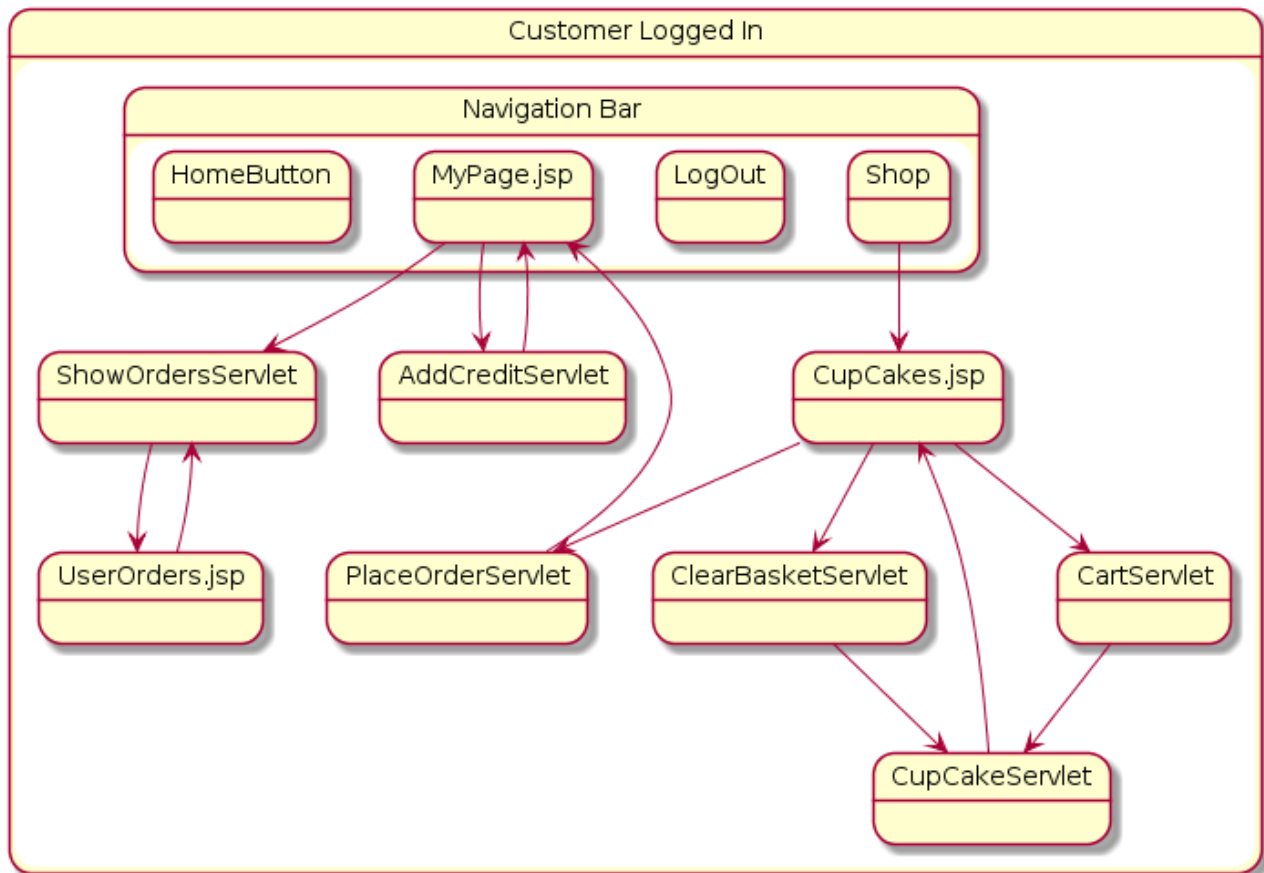
5. Navigationsdiagrammer

I dette afsnit optræder tre Composite State Diagrams, som tilsammen udgør en samlet visning af de servlets og jsp-filer der omfattes af iCupCake. Det første viser helt enkelt navigationen fra startsiden index.jsp og indtil kunden er logget ind, eventuelt efter at have registreret sig. Det næste viser navigationen for kunden som er logget ind og kan handle i shoppen, og endelig viser det tredje navigationen for admin-delen af siden.

Figur 5.1 Composite State Diagram for Login/Register:



Figur 5.2 Composite State diagram for kunde logget ind:



Her er det værd at bemærke at de øverste fire states repræsenterer en fælles navigationsbar:

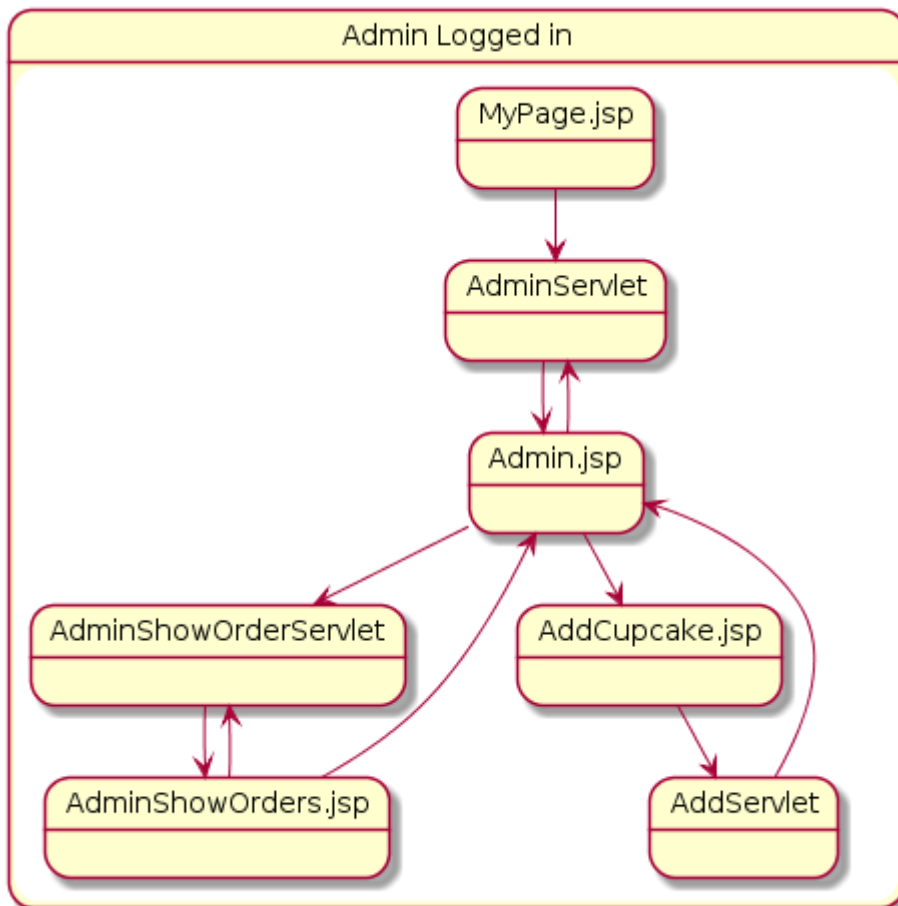
- HomeButton
- Shop
- MyPage
- Log Out

Således kan man som customer (og Admin) til enhver tid klikke på en af de fire for at bevæge sig til dette udgangspunkt. HomeButton er i denne sammenhæng Velkommen-siden, hvorfra man kan gå til login eller direkte til shoppen, som kan ses uden at man er logget ind – der kan dog ikke bestilles noget uden at man er logget ind.

Nedenfor vises Composite State Diagram for Admin logget ind. Her skal diagrammet ses i lyset af, at man som Admin i udgangspunktet ser samme side som customer lige efter

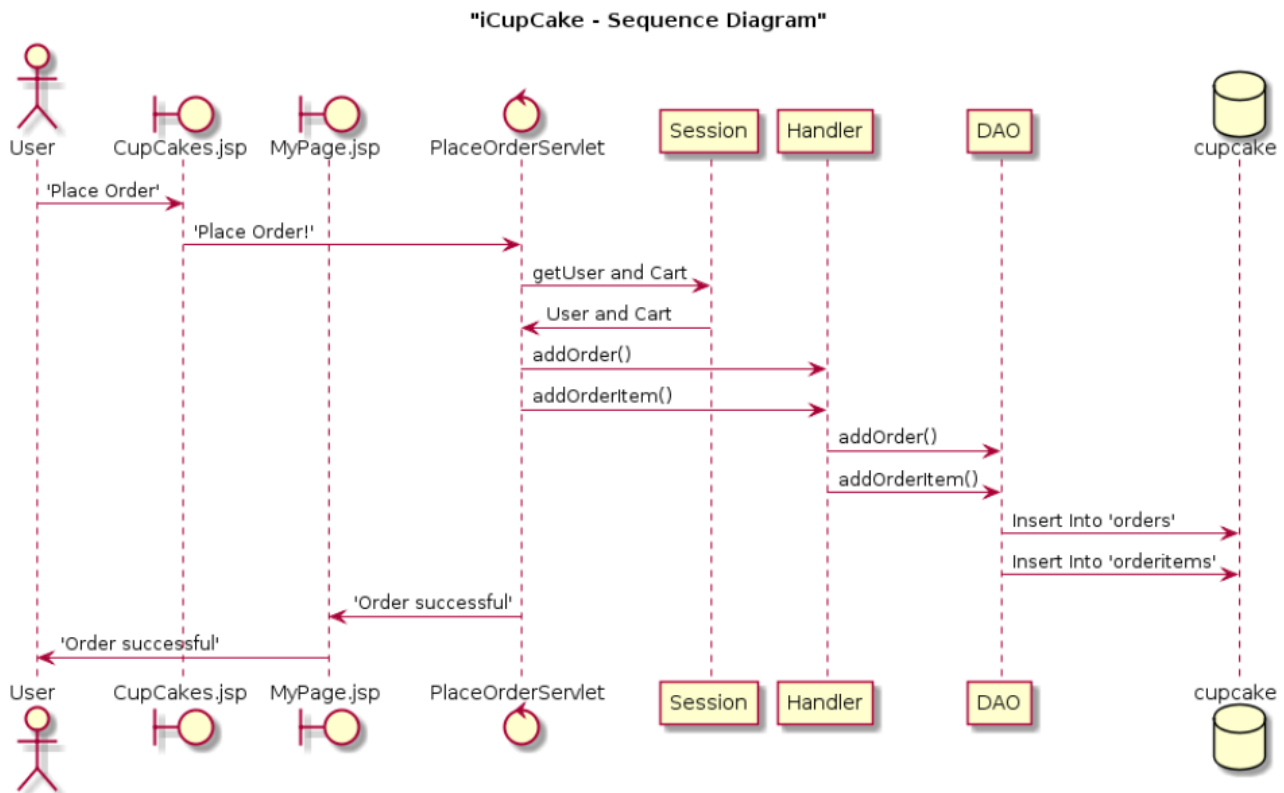
login. Man har så mulighed for, fra MyPage.jsp, at klikke på en admin knap der bringer én hen hvor man kan navigere mellem adminsiderne som beskrevet i diagrammet herunder.

Figur 5.3 Composite State Diagram for admin logget ind:



6. Sekvensdiagram for cupcake-køb:

Figur 6.1 Sekvensdiagram:



Dette sekvensdiagram beskriver flowet idet en bruger har valgt en cupcake og derefter trykker på 'Place Order'. PlaceOrderServlet henter User og ShoppingCart i sessionen, og kalder herefter metoderne addOrder() og addOrderItem() på Handler og DAO. Returtypen på disse metoder er void, hvorfor PlaceOrderServlet efter disse kald fortsætter med en requestForward til MyPage.jsp. Denne giver så brugeren besked om en succesfuld ordre.

MyPage.jsp kan også nås uden at man netop har placeret en ordre. Der er således en if-sætning i begyndelsen af MyPage.jsp, der evaluerer på om der ligger en attribut i requesten med navnet 'ordersuccess'. Hvis ja, udskrives teksten om at ordren var succesfuld, og man har derudover muligheder som vanligt på MyPage.jsp.

7. Særlige forhold

7.1 Session:

Informationer der gemmes i session er følgende: Den bruger der er logget ind, gemmes som et User objekt. Dette sikrer at man automatisk logges ud efter en periode med inaktivitet. Endvidere gemmes de cupcakes som kunden har lagt i indkøbskurven i en ArrayList af Cupcake objekter.

7.2 Validering af brugerinput:

- Ved indlogging: det checkes via metoden loginAuthentication at email og password matcher en eksisterende bruger i databasens user-tabel.
- Ved registrering af ny bruger: det checkes via metoden checkIfRegisterable at hverken det indtastede brugernavn eller password er en tom streng, og at den indtastede email ikke findes i forvejen i databasens user-tabel.
- Ved tilføjelse af credits til brugerens konto checkes det at inputtet er mellem 1 og 10000. Dette check foregår i AddCreditServlet.
- Ved 'shopping': Hvis man ikke er logget ind, eller hvis man har lagt mere i indkøbskurven end man har credits til at købe, er 'Place Order' knappen utilgængelig. Dette check foretages i CupCakes.jsp. Endvidere checkes det at man lægger minimum 1 og maximum 100 i kurven af hver enkelt cupcake af gangen – dette checkes i CartServlet.

Alle sådanne checks producerer en meddelelse til brugeren, hvis det pågældende check viser at der er foretaget et uregelmæssigt input.

7.3 Sikkerhed ved login:

Ved login videresendes inputs via post-metoden, der sikrer at loginoplysningerne ikke kan ses i browserens adressevindue.

8. Status på implementation

Vi har implementeret alt hvad opgaven krævede, alle JSP sider er funktionelle og stilet med bootstrap. Vi valgte at holde vores "Register.jsp" side i samme tema som resten af vores sider, og bootstrappe den, fremfor den brune baggrund beskrevet i opgave 9.4.