

CUPCAKERAPPORT

Projekt om at lave et website til en fiktiv cupcakebutik



Marie Fiskbæk

cph-mf322@cphbusiness.dk

github: mariefiskbak

April 2022

DATAMATIKER BORNHOLM 2. SEMESTER

INDHOLDFORTEGNELSE

INDHOLDFORTEGNELSE	1
INDLEDNING	2
Baggrund	2
Teknologivalg	2
KRAV	3
AKTIVITETSDIAGRAM	4
DOMÆNEMODEL OG EER-DIAGRAM	5
Domænemodel:	5
EER diagram	
Følgende EER-diagram viser hvordan databasen hænger sammen. Den er på 3. normalform.	6
NAVIGATIONSDIAGRAM	7
SÆRLIGE FORHOLD	8
STATUS PÅ IMPLEMENTATIONEN	8
PROCESS	9

INDLEDNING

Dette projekt handler om at forestille sig, at vi er blevet hyret til at levere en hjemmeside til salg af cupcakes i Olsker. Projektet tager bygger oven på en startkode, vi har fået udleveret.

Målgruppen for rapporten er en anden datamatikerstuderende på 2. semester eller person med tilsvarende forudsætninger, som med denne i hånden gerne skulle kunne forstå hvordan hjemmesiden er lavet.

Baggrund

Det er en lille nystartet virksomhed, Olsker Cupcakes, der har bestilt en hjemmeside. De laver cupcakes, som man kan afhente i butikken. Man sammensætter selv sin cupcake ved at vælge bund og topping.

Kunderne skal kunne oprette en profil med konto på hjemmesiden og bestille cupcakes med valgfri top og bund.

Butikken skal kunne logge på som administrator og se alle ordrer der ligger.

Teknologivalg

Der er brugt følgende software til projektet:

Apache Tomcat 9.0.60

HikariCP 4.0.3

IntelliJ 2021.2.1

Java 1.8.0_282

Maven 3.3.2

MySQL Workbench 8.0.28

Twitter BOOTstrap 5.1.3

KRAV

Virksomhedens ønske er, at gøre det nemt for kunderne at bestille cupcakes og nemt for virksomheden at få overblik over hvad der er bestilt, og hvad der skal laves.

Det overordnede mål er delt op i en række mindre *user stories*:

US-1: Som kunde kan jeg bestille og betale cupcakes med en valgfri bund og top, sådan at jeg senere kan køre forbi butikken i Olsker og hente min ordre.

US-2: Som kunde kan jeg oprette en konto/profil for at kunne betale og gemme en ordre.

US-3: Som administrator kan jeg indsætte beløb på en kundes konto direkte i MySQL, så en kunde kan betale for sine ordrer.

US-4: Som kunde kan jeg se mine valgte ordrelinier i en indkøbskurv, så jeg kan se den samlede pris.

US-5: Som kunde eller administrator kan jeg logge på systemet med email og kodeord. Når jeg er logget på, skal jeg kunne se min email på hver side (evt. i topmenuen, som vist på mockup'en).

US-6: Som administrator kan jeg se alle ordrer i systemet, så jeg kan se hvad der er blevet bestilt.

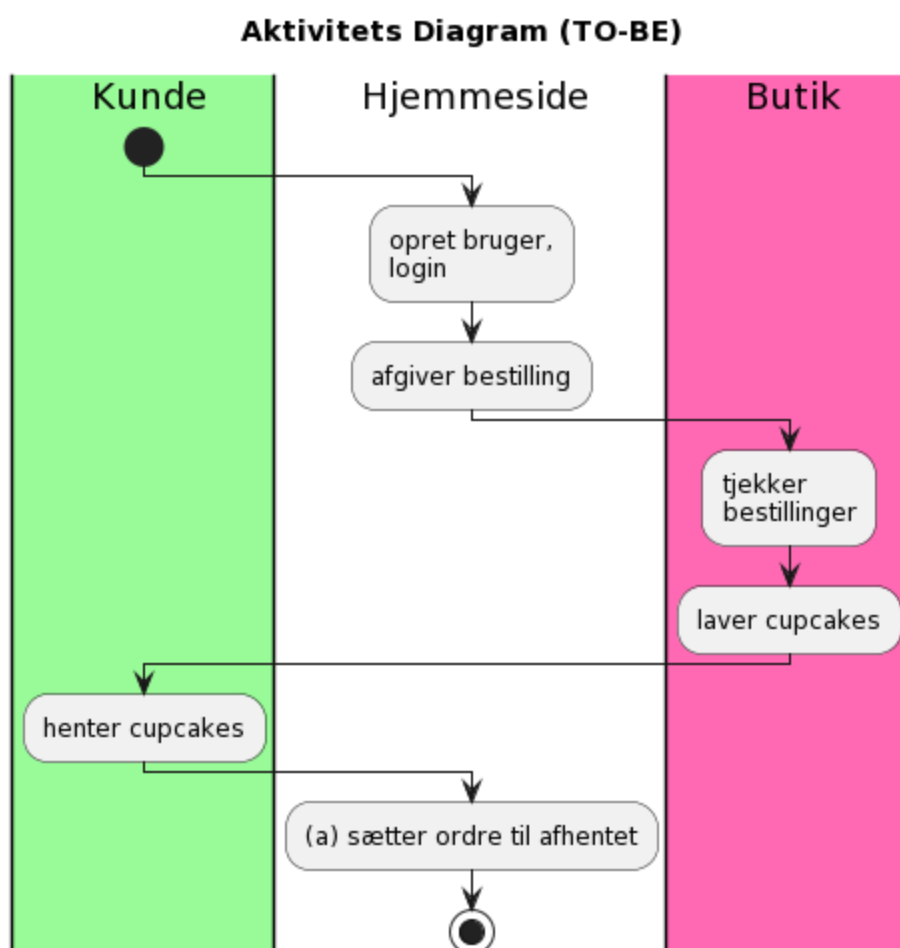
US-7: Som administrator kan jeg se alle kunder i systemet og deres ordrer, sådan at jeg kan følge op på ordrer og holde styr på mine kunder.

US-8: Som kunde kan jeg fjerne en ordre fra min indkøbskurv, så jeg kan justere min ordre.

US-9: Som administrator kan jeg fjerne en ordre, så systemet ikke kommer til at indeholde ugyldige ordrer. F.eks. hvis kunden aldrig har betalt.

AKTIVITETSDIAGRAM

Et aktivitetsdiagram er nyttigt til at få overblik over arbejdsgangen i virksomheden. Her er lavet et lille et for at vise det flow, der skal være, når hjemmesiden er implementeret. (a) står for admin.



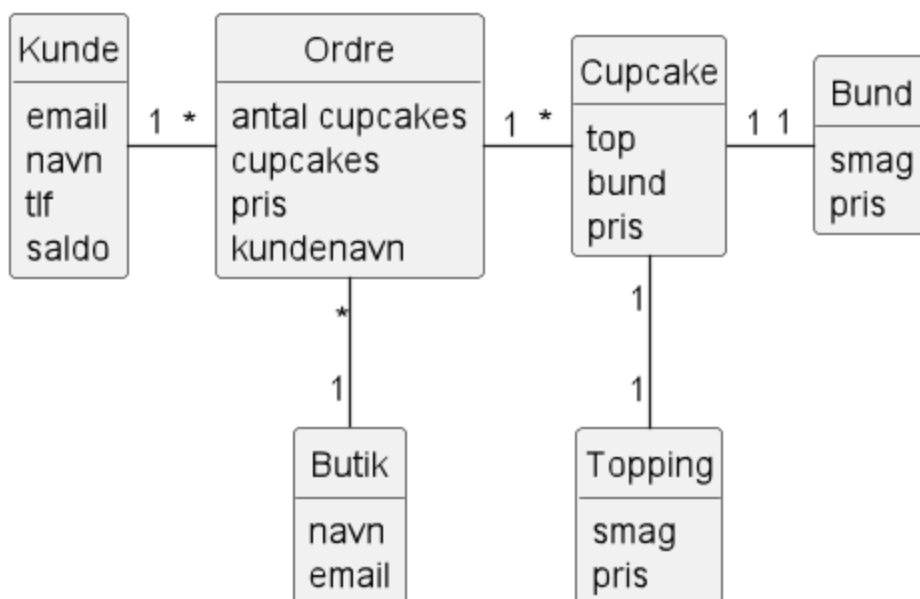
DOMÆNEMODEL OG EER-DIAGRAM

Jeg har lavet en domænemodel for at give overblik over hvad det system jeg har lavet skal håndtere og et EER (enhanced entity-relationship diagram) for at vise hvordan databasen, som systemet henter fra og skriver til hænger sammen.

Domænemodel:

En cupcake består af en valgfri top og bund, som har fået hver sin kasse. En ordre kan indeholde flere cupcakes, en kunde kan lave flere ordre og butikken kan have mange ordre. Ordren er således forbindelsesleddet mellem butikken og kunden.

Domænemodel Olsker Cupcakes



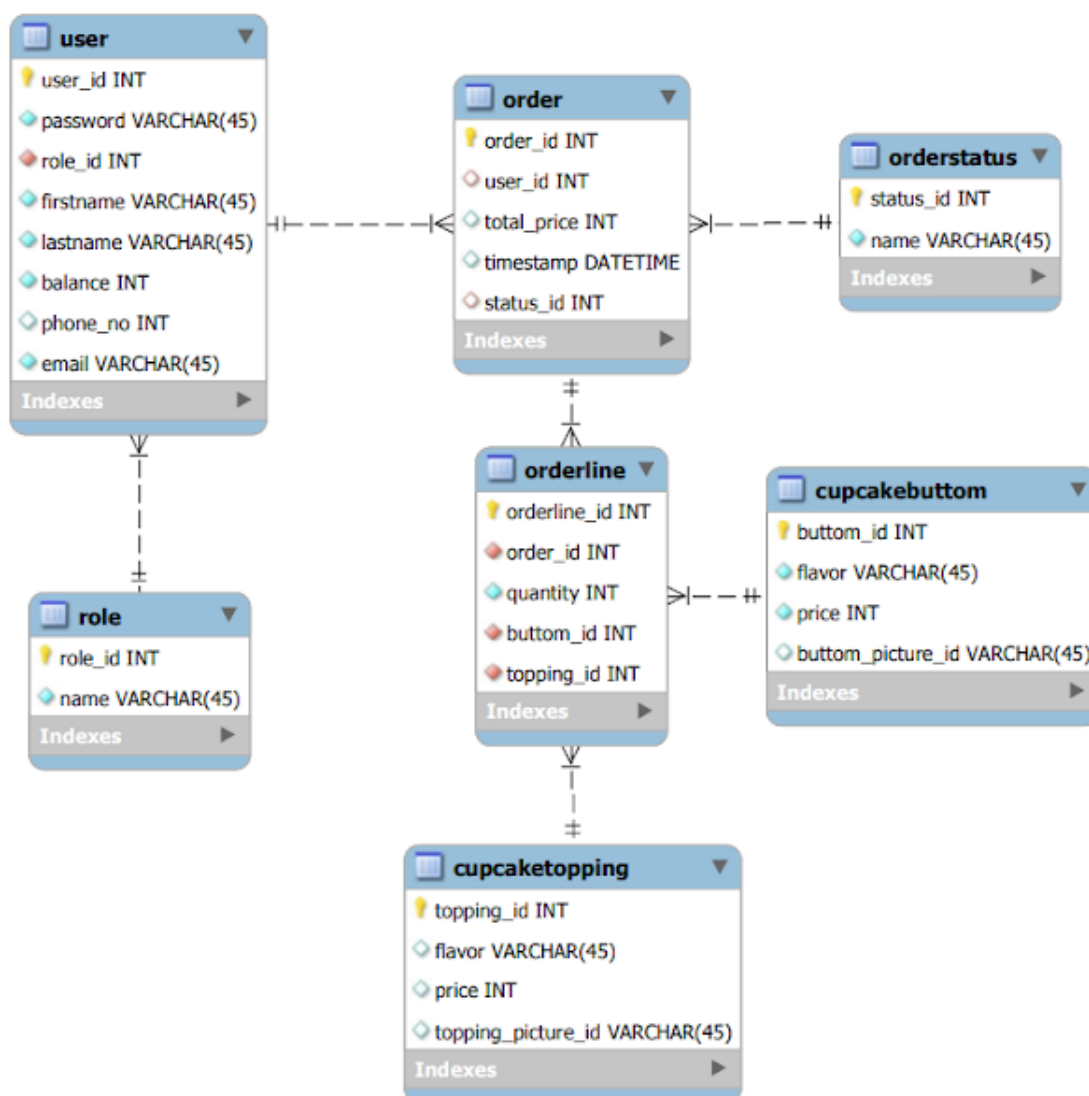
EER diagram

Følgende EER-diagram viser hvordan databasen hænger sammen. Den er på 3. normalform.

Alle tabellerne har deres eget id som primær nøgle. Order-id'et er sat til Auto Increment, og bliver således genereret automatisk. Orderlinje-id'et er en logisk nøgle og består af ordre-id'et efterfulgt af bund-id'et og topping-id'et. Det er valgt så id'et kan sendes med fra knapperne i indkøbskurven, da jeg gerne vil samle alle ordrelinjer, med den samme cupcake til en linje, og blot opdatere antallet.

Jeg bruger ikke picture-id'erne under cupcakebutton og cupcaketopping i dette projekt, de er et levn fra gruppeprojektet, jeg endte med at gå ud af.

Fremmednøglerne er valgt for at kunne forbinde tabellerne med hinanden. Der er kun brugt Not Null constraint, og det er for at sikre at der ikke kommer mangelfuld data ind i tabellen.

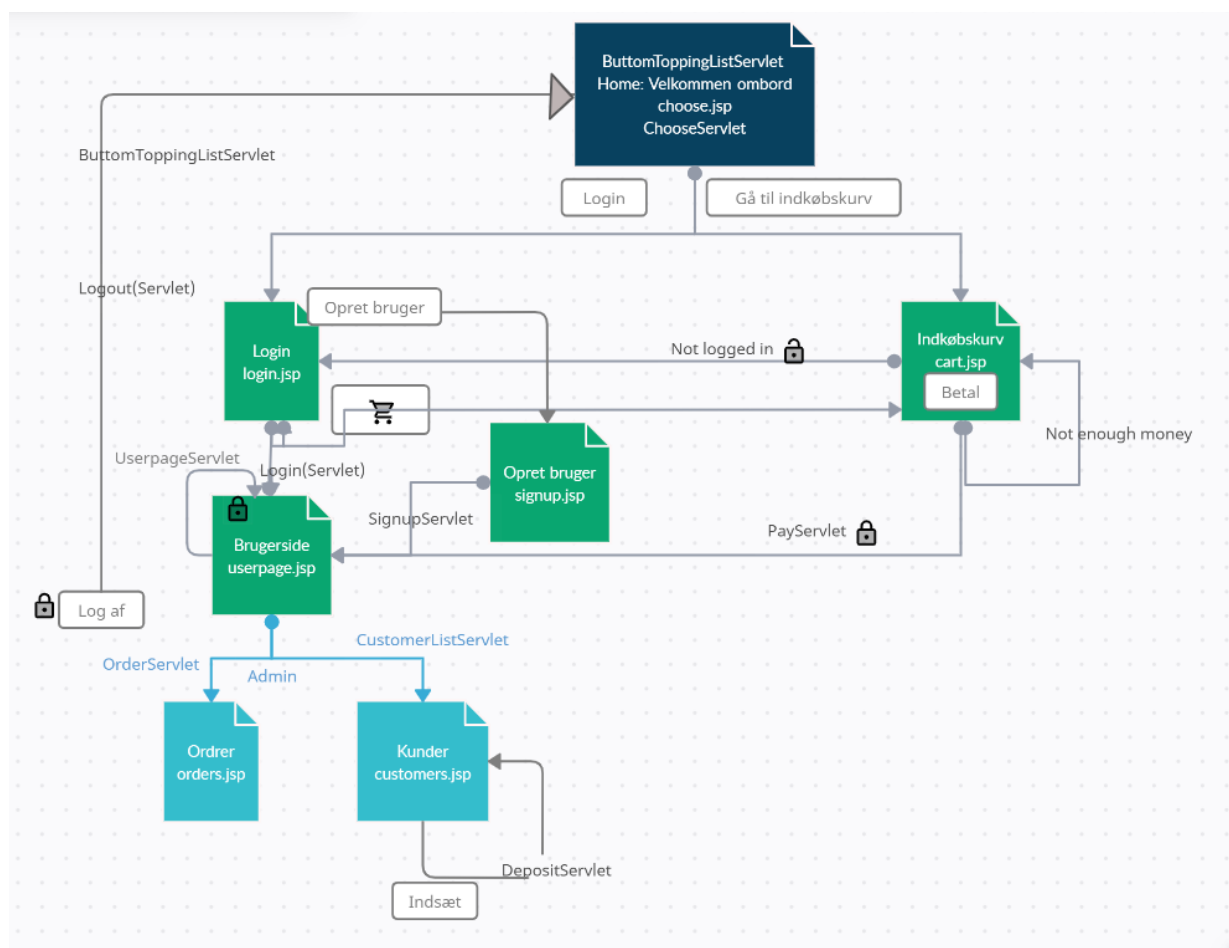


NAVIGATIONS DIAGRAM

Websitet er en navigationsbar som indeholder ganske få menupunkter, som her:

IKKE LOGGET PÅ	LOGGET PÅ SOM BRUGER	LOGGET PÅ SOM ADMIN
Login	Min side	Ordre
		Kunder
	Log af	Log af
	E-mail (brugerside)	E-mail (brugerside)
Indkøbskurv	Indkøbskurv	Indkøbskurv

Navigationsdiagrammet her skal vise hvordan websitets sider hænger sammen og hvilke servlets der bruges, samt navnene på jsp-siderne. Knapper er illustreret med en hvid boks med samme tekst i, som på websitet. De blå sider er det kun admin der kan tilgå (se dog afsnittet om status på implementationen). Man skal desuden være logget ind for at bestille kager og for at se sin ordrehistorik.



SÆRLIGE FORHOLD

Følgende emner gemmes i Sessionscope:

orderId, cartDTolist, totalpris, totalantalcupcakes, orderlineDTomap, customerlist, user, orderlist, orderlinelist, orderlinelistmap, listoflists, userorderlist.

Buttomlist og toppinglist gemmes på requestscopet.

Jeg har ikke prioriteret at bruge tid på exceptions, og det er således hovedsagligt dem der følger med fra startkoden, samt IntelliJ's autogenererede der er bruges.

Når der oprettes en bruger bedes der om at password skal indtastes to gange, og der tjekkes om de stemmer overens. Det tjekkes også om e-mailadressen allerede findes i databasen. Ellers valideres brugerinput ikke andet end det inputfelterne og databasen selv sørger for.

For at øge sikkerheden i forbindelse med login, er det doPost-metoden, der bruges, så det indtastede ikke vises i URL'en, men informationen krypteres ikke. Det ville være spændende at sætte kryptering på en anden gang.

STATUS PÅ IMPLEMENTATIONEN

Alt i alt er jeg tilfreds med der hvor jeg er nået til i projektet, men der er selvfølgelig en del ting jeg ikke nåede alligevel.

- Der er ikke sat en knap ind til at ændre brugernes status til admin.
- Der er ikke lavet en knap hvor kunden kan slette hele sin igangværende bestilling på een gang.
- Ville have været fedt at samle administratorens kunde- og ordretabel til én og have forskellige sorterings og søgemuligheder.
- Når en bruger logger ind, printes tidligere ordrer ikke med det samme på siden, da servletten simpelthen ikke ville kalde en anden servlet (det vil den ellers gerne et andet sted i koden). Så man skal lige klikke på "min side" før det kommer frem.
- Fejlbeskeder er der ikke gjort så meget ud af og det er ikke alle der virker.
- Unittest har jeg ikke brugt.
- Koden er ikke ryddet op og refaktoreret.

- Sider der er forbeholdt indloggede brugere, er ikke sikrede, hvis brugeren skriver URL'en direkte i browseren, så fx kan man se kundelisten uden at være logget på som admin ved at skrive http://localhost:8080/nycupcakemarie_war_exploded/CustomerListServlet direkte i URL'en. Eneste sikkerhed pt er at de knapper der leder hen til siden, først dukker op, når man er logget ind.

PROCESS

Jeg synes 2. semester har været lidt for "huhej" til at være lærerigt, så det var denne gang vigtigere for mig at virkelig få lært at bruge HTML, CSS, Bootstrap og Servlets end at prioritere gruppearbejdet, så da jeg ikke syntes det gled, lavede jeg så mit eget projekt.

Det havde dog nok været godt at komme i gang med at bruge git bedre fra start af, vi fandt først ud af hvordan man laver branches osv efter en lille uge.

Jeg kunne godt tænke mig at have øvet mig i at bruge Unit tests aktivt, men blev skåret fra af hensyn til arbejdstiden og at der allerede var mange nye ting at lære.