

OLSKER CUPCAKES

Datamatiker – 2. semester – E-klassen

Forår 2020

HASSANAIN ALI

cph-ha170@cphbusiness.dk

Github:

ha-170

<https://github.com/ha-170>

ANDREAS M. A. ANDERSEN

cph-aa344@cphbusiness.dk

Github:

etlos01

<https://github.com/Etlos01/>

LINK TIL AFLEVERING PÅ GITHUB

<https://github.com/Etlos01/Cupcakes>

DEADLINE FOR AFLEVERING AF PROJEKT

Søndag d. 29. marts 2020 kl. 23:55



CPH BUSINESS LYNGBY

Indholdsfortegnelse

INDLEDNING OG BAGGRUND	1
TEKNOLOGIVALG.....	1
FUNKTIONELLE KRAV	2
EER-DIAGRAM.....	2
SEKVENSDIAGRAM.....	3
SÆRLIGE FORHOLD OG STATUS PÅ IMPLEMENTATION	4

Indledning og baggrund

Olsker Cupcakes er et lille bageri på Bornholm, som på det seneste har oplevet en stor vækst og efterspørgsel på Deres cupcakes. I den forbindelse har virksomheden et ønske om et system, der kan gøre det nemmere at varetage og håndtere de mange bestillinger.

Virksomheden ønsker at kunden skal kunne bestille og betale for sin cupcakes med valgfri bund og top, så kunden senere kan afhente sin ordre. Kunden skal kunne oprette en brugerprofil for at kunne betale og gemme sine ordrer, og kunden skal kunne se sine ordrelinier i en indkøbskurv med den samlede pris. Sidst men ikke mindst skal kunden naturligvis også kunne slette en ordrelinie fra indkøbskurven.

Virksomheden ønsker at kunne oprette administratorkontoer for sine medarbejdere. Som administrator skal man kunne indsætte et beløb på en kundes konto, så kunden kan betale for sine bestillinger. Administrator skal kunne se alle ordrer i systemet og også kunne slette ordrer, hvis kunden ikke får betalt sin ordre. For både medarbejdere og kunder gælder det, at man kan logge på systemet ved at gøre brug af en e-mail og kodeord, og man skal kunne se sin e-mail på hver side.

Denne rapport omfatter dokumentation af udarbejdningen af det ønskede system, der skal håndtere bestillinger af cupcakes for virksomheden, Olsker Cupcakes.

Teknologivalg

Der er anvendt følgende teknologier til udvikling af cupcake-projektet:

- ◇ IntelliJ IDEA Ultimate Edition 2019.3.4
- ◇ Tomcat 7.0.103
- ◇ MySQL Workbench 8.0.19
- ◇ Maven
- ◇ Java
- ◇ HTML
- ◇ JavaScript

Funktionelle krav

Som der også nogenlunde er beskrevet under afsnittet Indledning og baggrund, ønsker kunden at få implementeret følgende tiltag/user-stories i produktet:

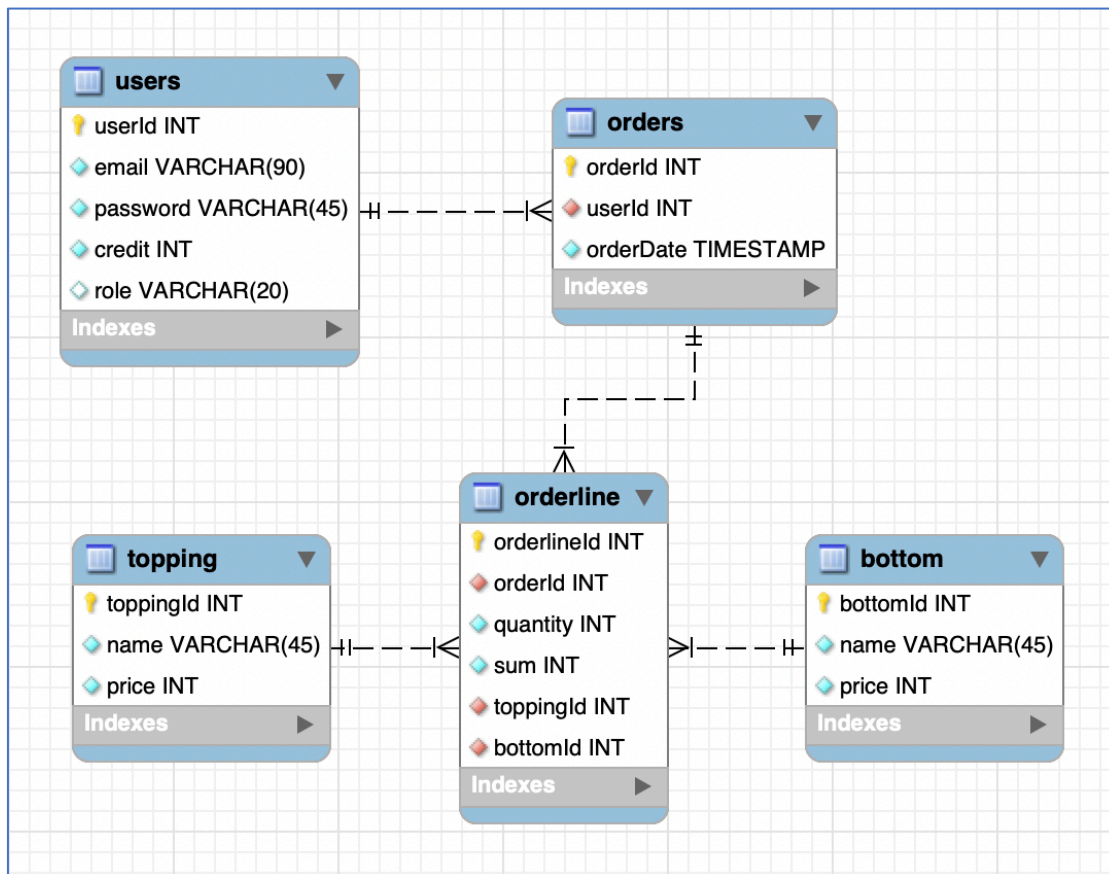
1. Som kunde kan jeg bestille og betale cupcakes med en valgfri bund og top, sådan at jeg senere kan køre forbi butikken i Olsker og hente min ordre.
2. Som kunde kan jeg oprette en konto/profil for at kunne betale og gemme en ordre.
3. Som administrator kan jeg indsætte beløb på en kundes konto direkte i MySQL, så en kunde kan betale for sine ordrer.
4. kunde kan jeg se mine valgte ordrelinier i en indkøbskurv, så jeg kan se den samlede pris.
5. Som kunde eller administrator kan jeg logge på systemet med e-mail og kodeord. Når jeg er logget på, skal jeg kunne se min e-mail på hver side.
6. Som administrator kan jeg se alle ordrer i systemet, så jeg kan se, hvad der er blevet bestilt.
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.
8. Som kunde kan jeg fjerne en ordre fra min indkøbskurv, så jeg kan justere min ordre.
9. Som administrator kan jeg fjerne en ordre, så systemet ikke kommer til at indeholde ugyldige ordrer f.eks. hvis kunden aldrig har betalt.

EER-diagram

Figur 1 viser et udvidet relationsdiagram over databasen, som er autogenereret via MySQL Workbench.

Tabellerne er på 3. normalform, og de har alle 1 til mange relationer. Det vil sige, at en kunde fra *user-tabellen* godt kan have flere ordrer i *orders-tabellen*, og en ordre kan godt have flere ordrelinier i *orderlines-tabellen*.

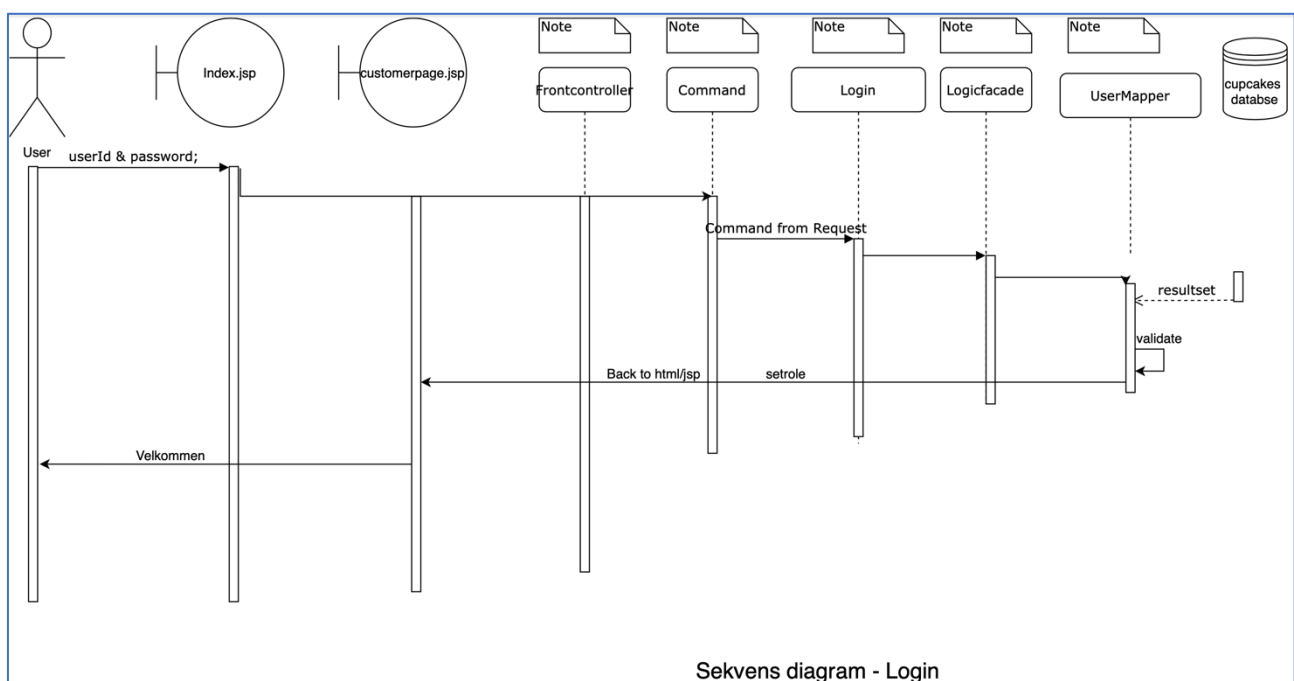
Samtlige tabeller har et unikt *primary key*, som er et ID, hvor der er sat auto-increment på, så hver gang der oprettes en ny bruger, ordre eller ordrelinie. *userId* er *foreign key* i *orders-tabellen*. Orderline har tre fremmednøgler, som er primærnøglerne fra tabellerne *orders*, *topping* og *bottom*. Alle fremmednøgler i tabellerne er oprettet som *restrict on update & delete*. Det giver databasen en form for sikkerhed i databasen, da man fx ikke kan oprette en ordre, hvis kunden ikke findes. Omvendt kan en bruger ikke slettes, hvis der er en aktiv ordre.



Figur 1 Enhanced Entity Relationship diagram for databasen.

Sekvensdiagram

På figur 2 ses et sekvensdiagram for Login.



Sekvens diagram - Login

Figur 2 Sekvensdiagram for Login

Særlige forhold og status på implementation

I programmet gemmes der nogle informationer i sessions, som er userId, role, email, bottomList og toppingList. De sidstnævnte to er ArrayLists, som er oprettede for at hente cupcaketoppe og -bunde fra databasen.

I programmet er der ikke implementeret nogen særlig form for sikkerhed i forbindelse med login eller i forhold til validering af brugerinput. Man kan oprette en konto ved at skrive noget vilkårligt i stedet for en e-mail. Sent opstart af projekt dermed tidspres var medskyld i, at vi ikke fik implementeret dette.

Vi burde lave noget sikkerhed, så en bruger kun kan oprette en konto ved at bruge en email, hvor der skal være snabel-a, som skal være afsluttet med et domæne, et punktum og fx "com". Man skulle heller ikke kunne oprette en bruger med samme e-mail flere gange, hvilket man på nuværende tidspunkt kan. På den måde ville man have undgået mærkelige formater og dupletter.

I forbindelse med password til login, kunne vi i forbindelse med oprettelse have sat krav til det, så det fx minimum skulle være med otte karakterer langt, både store og små bogstaver samt tal og specialtegn. Ydermere kunne man sætte en spærring på, hvis koden til en konto gættes forkert fx fem gange i træk.

Pga. tidspres blev mockup'en sprunget over før vi var halvvejs færdig med den. I begyndelsen rodede vi med CSS og bootstrap samtidig med userstory 1, og det udskød vi, da det tog langt tid, så ville vi vente med at få det implementeret. Userstories 1 til 5 er færdige, men vi har ikke nået at kigge videre på CSS/Bootstrap. Vi mangler en del diagrammer og må konkludere, at dette har været en læringsproces og træning til eksamensprojektet.