

# Olsker Cupcakes

## Datamatikeruddannelsen

Gruppe 1, hold 20dat2e

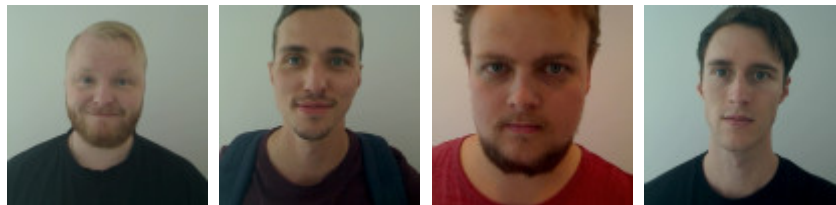
### Gruppemedlemmer:

Alex Richardt Gottorp Wagner (GH: A-R-G Wagner)  
(cph-aw116@cphbusiness.dk)

Benjamin Choleva (GH: KingBendico)  
(cph-bc105@cphbusiness.dk)

Joakim Stensnæs (GH: JoakimKSS)  
(cph-js437@cphbusiness.dk)

Lasse Emil Støvring Larsen (GH: Lforlasse)  
(cph-ll330@cphbusiness.dk)



Copenhagen Business Academy  
30. marts 2020

## **Abstract**

This report documents our first second semester Datamatiker-project at Copenhagen Business Academy. We were tasked with designing and creating a webshop for a cupcake store.

# Indhold

<b>1</b>	<b>Indledning</b>	<b>3</b>
1.1	Formål . . . . .	3
1.2	Baggrund . . . . .	3
1.3	Teknologivalg . . . . .	4
<b>2</b>	<b>Krav</b>	<b>5</b>
2.1	Bageriet Olsker Cupcakes . . . . .	5
2.2	User stories . . . . .	5
<b>3</b>	<b>Diagrammer</b>	<b>6</b>
3.1	ER-Diagram . . . . .	6
3.2	Sekvensdiagram . . . . .	7
3.3	UML-Diagram . . . . .	8
3.4	Navigationsdiagram . . . . .	9
3.5	Domænediagram . . . . .	10
<b>4</b>	<b>Særlige forhold</b>	<b>11</b>
<b>5</b>	<b>Status på implementation</b>	<b>12</b>
<b>6</b>	<b>Konklusion</b>	<b>13</b>

# 1 Indledning

## 1.1 Formål

Vi skulle lave en webstore til bageriet Olsker Cupcakes. Vi havde fået 17 dage, fra d. 12. marts til d. 29. marts 2020, til at udvikle en webstore til bageriet Olsker Cupcakes på Bornholm.

## 1.2 Baggrund

Vi har landet en vigtig opgave fra Olsker Cupcakes. Det er endnu et ”dybdeøkologisk iværksættelseventyr” fra Bornholm, som har ramt den helt rigtige opskrift. Et par hipstere fra København har været forbi bageriet, indsamlet nogle krav og lavet en halvfærdig mock-up af en tænkt forside.

De havde følgende krav:

En kunde skal kunne bestille og betale cupcakes med en valgfri bund og top, sådan at de senere kan køre forbi butikken i Olsker og hente deres ordre. Denne skal også kunne oprette en konto/profil for at kunne betale og gemme sin ordre.

En administrator skal kunne indsætte et beløb på en kundes konto, så denne kan betale for sine ordrer. En kunde skal kunne se sine valgte ordrelinjer i en indkøbskurv og kunne se den samlede pris.

En kunde eller administrator skal kunne logge på systemet med e-mail og adgangskode. Når de er logget på, skal de kunne se deres e-mail på hver side.

En administrator skal kunne se alle ordrer i systemet, så denne kan se, hvad der er blevet bestilt.

En administrator skal kunne se alle kunder i systemet og deres ordrer, sådan at denne kan følge op på ordrer og holde styr på sine kunder.

En kunde skal kunne fjerne en ordre fra sin indkøbskurv, så denne kan justere sin ordre.

En administrator skal kunne fjerne en ordre, så systemet ikke kommer til at indeholde ugyldige ordrer. F.eks. hvis kunden aldrig har betalt.

### 1.3 Teknologivalg

- IntelliJ IDEA 2019.3.3
- JDBC
- MySQL Workbench 8.0 CE
- MySQL 8.0.18 (2019-10-14)
- Adobe XD
- apache-tomcat-8.5.53.
- DigitalOcean Webserver
- Ubuntu 18.04.3 (LTS) x64

## 2 Krav

### 2.1 Bageriet Olsker Cupcakes

En webstore er med til at streamline en forretning. En Webstore giver kunder mulighed for at handle i døgnets 24 timer og er meget bekvem for kunderne. Derudover, så er der mulighed for at sælge til flere på kortere tid.

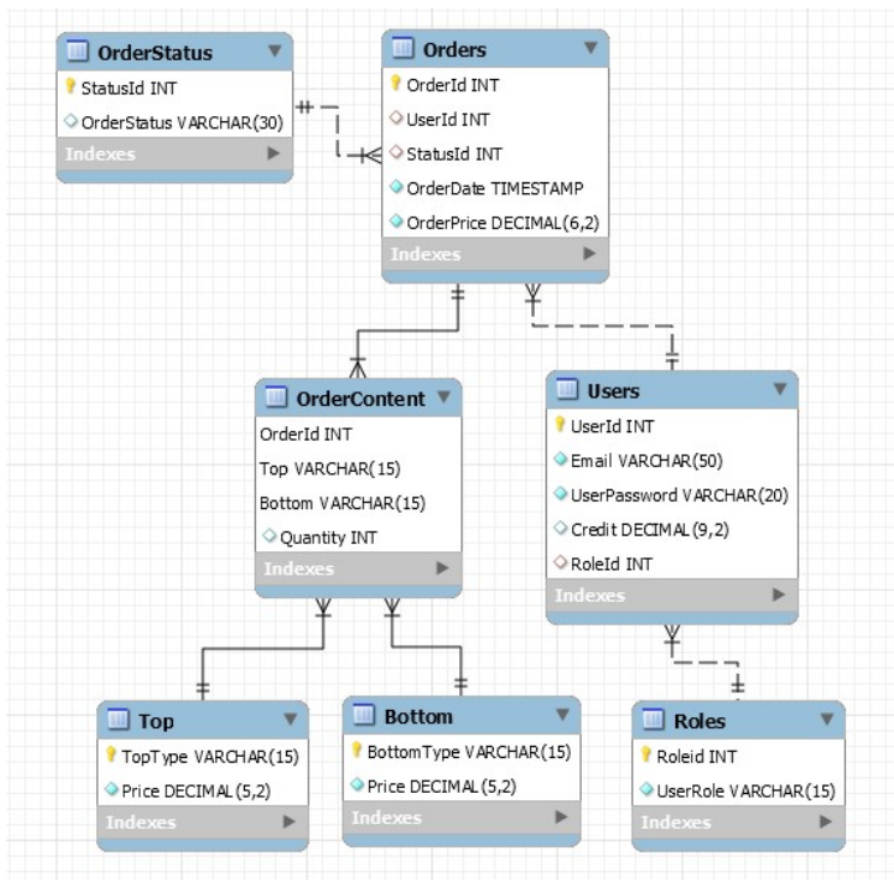
### 2.2 User stories

Vores opgave var at få udviklet en webstore med følgende krav:

- **US-1:** En kunde skal kunne bestille og betale cupcakes med en valgfri bund og top, sådan at kunden senere kan køre forbi butikken i Olsker og hente sin ordre.
- **US-2:** En kunde skal kunne oprette en konto/profil for at kunne betale og gemme en ordre.
- **US-3:** En administrator skal kunne indsætte et beløb på en kundes konto direkte i MySQL, så en kunde kan betale for sine ordrer.
- **US-4:** En kunde skal kunne se sine valgte ordrelinjer i en indkøbskurv og kunne se den samlede pris.
- **US-5:** En kunde eller administrator skal kunne logge på systemet med e-mail og kodeord. Når de er logget på, skal de kunne se deres e-mail på hver side.
- **US-6:** En administrator skal kunne se alle ordrer i systemet, så denne kan se, hvad der er blevet bestilt.
- **US-7:** En administrator skal kunne se alle kunder i systemet og deres ordrer, sådan at denne kan følge op på ordrer og holde styr på sine kunder.
- **US-8:** En kunde skal kunne fjerne en ordre fra sin indkøbskurv, så denne kan justere sin ordre.
- **US-9:** En administrator skal kunne fjerne en ordre, så systemet ikke kommer til at indeholde ugyldige ordrer. F.eks. hvis kunden aldrig har betalt.

## 3 Diagrammer

### 3.1 ER-Diagram



De to vigtigste dele af vores database er vores Orders tabel og vores Top og Bottom tabeller.

Top og Bottom tabellerne giver virksomheden mulighed for let at tilføje eller fjerne bunde og toppe i deres sortiment, samt at ændre prisen på de forskellige enheder.

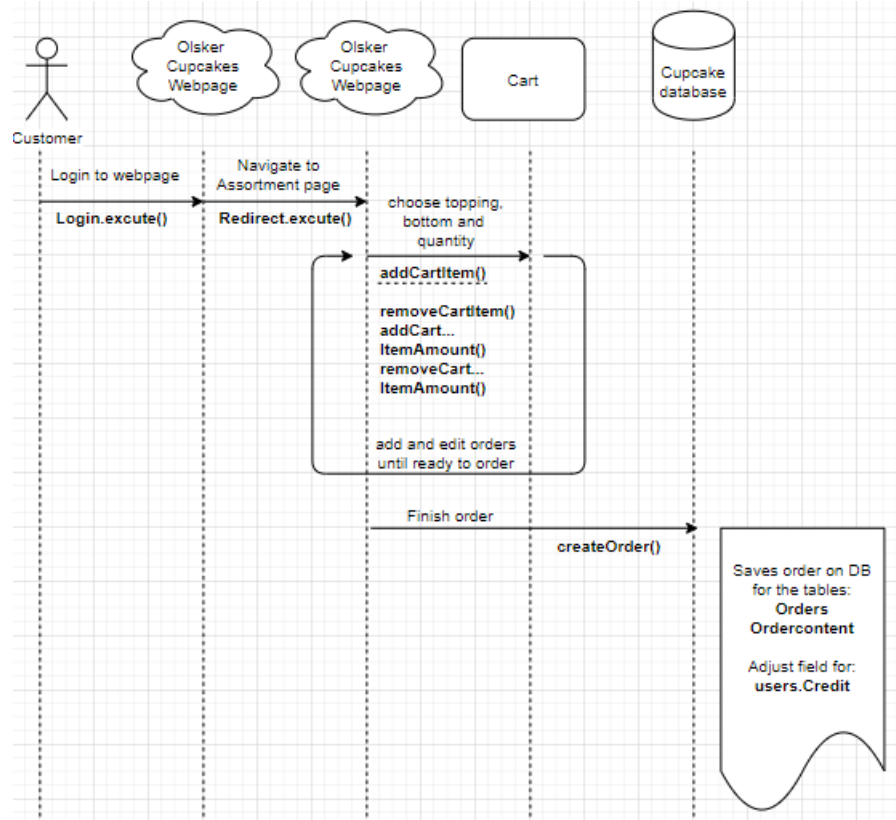
Orders indeholder en oversigt over alle ordrer som er kørt igennem systemet og kategoriserer hvor de er i bestillings processen, inklusive annullerede ordrer og gennemførte ordrer.

Vi bruger tabellerne Roles og OrderStatus, for at undgå deletion anomaly og bruger foreign keys til at refererer til dem.

Vi bruger foreign keys i OrderContent og Orders tabellerne, for at sikre at de personer, ordrer og item typer som refereres til, findes.

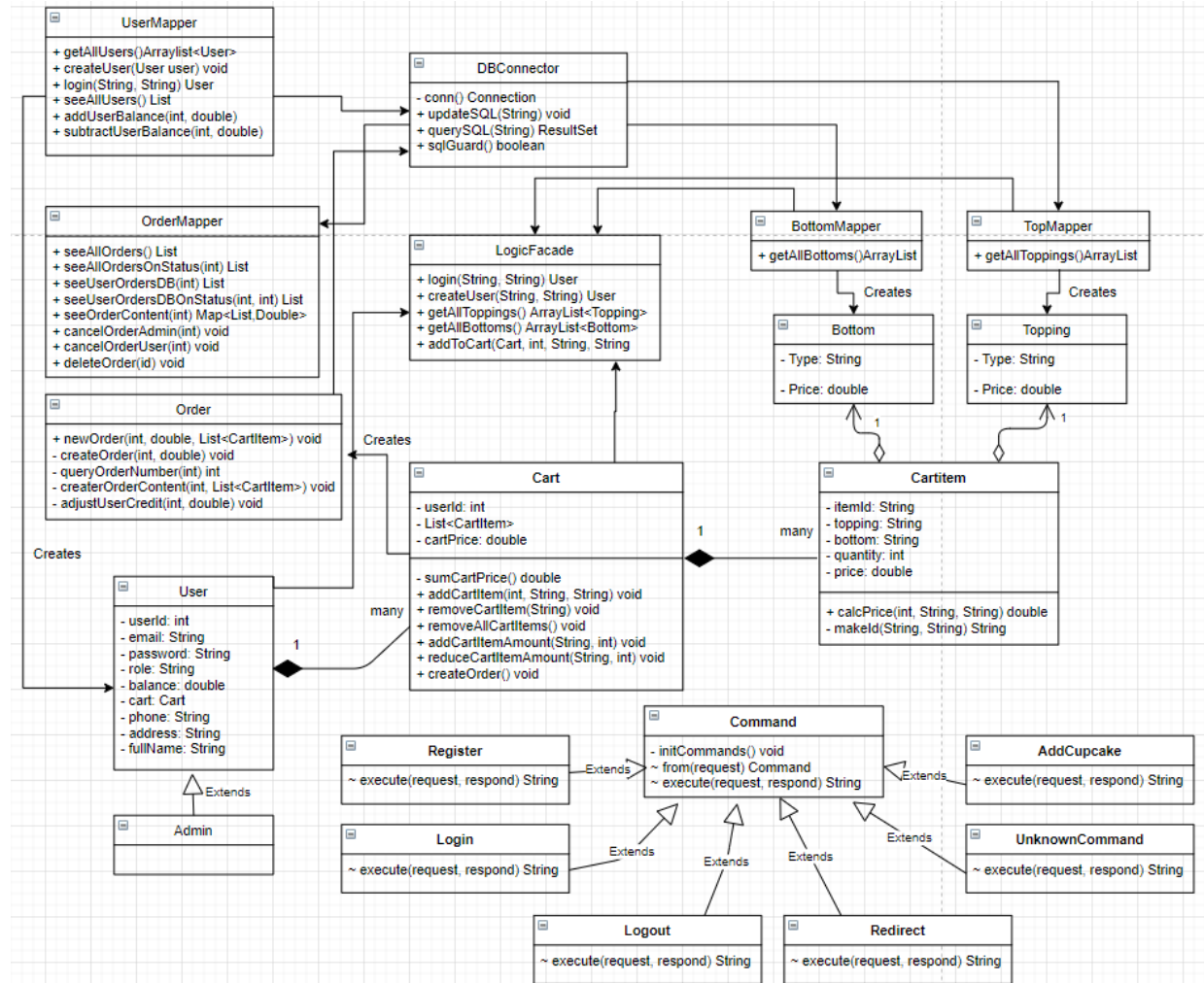
Ændringsforslag til projektet skulle være at OrderContent tabellen også indeholder prisen på den ordrelinie som den repræsenterer.

### 3.2 Sekvensdiagram

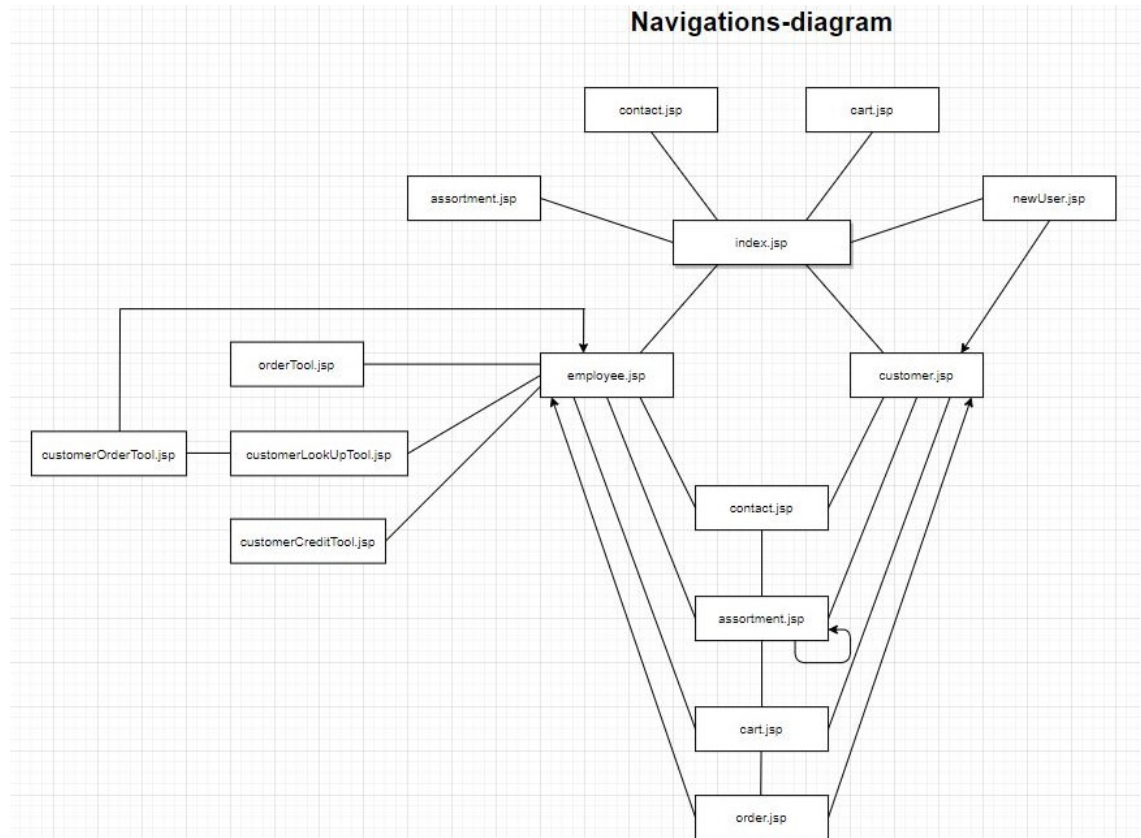




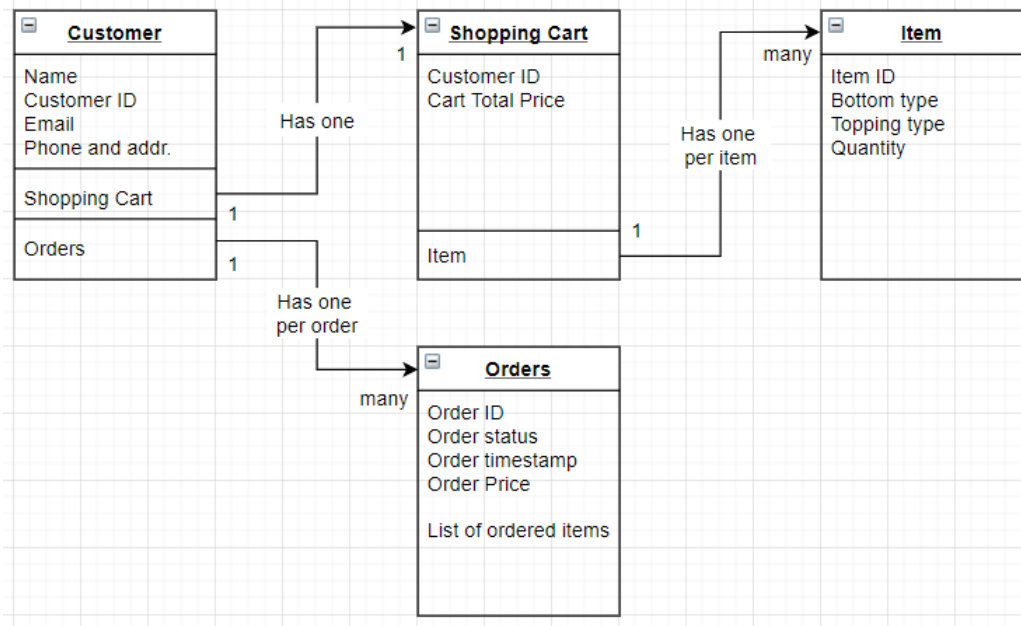
### 3.3 UML-Diagram



### 3.4 Navigationsdiagram



### 3.5 Domænediagram



## 4 Særlige forhold

Når en bruger benytter login-formularen bliver adgangskoden ikke vist i feltet. Adgangskoden bliver ikke krypteret / hashet. Hvis adgangskoden ikke stemmer overens med den indtastede e-mailadresse, så vil brugeren blive sendt tilbage til ‘index.jsp’ med en fejlkode. Ved brugerlogin, oprettes en ny session.

I denne session bliver alle de data, som tilhører en specifik bruger, gemt. Der bliver gemt e-mail, navn, rolle, telefonnummer og adresse.

Der bliver derudover også oprettet et ‘User’-objekt, som bliver gemt i den konkrete session. Når et stykke data er blevet benyttet, modificeret og gemt i databasen, bliver det hentet på ny og opdateret i den aktive session for den individuelle bruger.

Rollen for den bruger der er logget ind, bliver tjekket på de individuelle ‘.jsp’-sider. Såfremt at brugerens rolle ikke er tilladt på en given side, bliver brugeren henvist til ‘index.jsp’-siden.

Derudover har vi valgt udelukkende at skrive på engelsk for at undgå en mærkelig blanding af dansk og engelsk. Dog har vi valgt at indeholdet på webstoren skal være på dansk.

## 5 Status på implementation

Brugers kredit/balance bliver kun opdateret i session, når denne logger ind. Det vil sige, at hvis man bekræfter en ordre, så bliver balancen altså registreret korrekt i databasen, men bliver ikke ført tilbage til sessionen korrekt.

Når man fjerner et element fra sin kurv, så opstår der en null-pointer-exception, men elementet bliver stadigvæk fjernet.

Dette bevirker ligeledes, at den viste pris på kurven ikke stemmer overens med den faktiske pris.

Hvis man tilføjer et nyt element i kurven, bliver prisen opdateret korrekt i sessionen.

Denne fejl har ingen indflydelse på registreringen af ordren, med de korrekte data, i databasen.

## **6 Konklusion**

Vi har opfyldt minimumskravene ...