
# Cupcake

Hold B

**Marwa**
Cph-me319@cphbusiness.dk
github: Marwamnr

**Allan**
cph-ac326@cphbusiness.dk
github: AllanChandler

**Marlene**
cph-mb910@cphbusiness.dk
github: MB105

Dato: 10. april 2024

# Indholdsfortegnelse

# Indledning

Projektet består af en webapplikation, som er bygget i Java, html og css, projektet er også tilkoblet en PostgreSQL server. Opgaven lyder på at lave en cupcake-butik som skal blive tilgængeligt online (lokalt). Vi startede med først at lave en domænemodel, og ud fra vores domænemodel lavede vi en ERD i databasen, ud fra det prøvede vi at finde overensstemmelse med de funktionelle krav som opgaven besidder. Løbende har vi haft meget fokus på at planlægge, diskutere og holde hinanden opdateret, så vi trinvist kunne få udarbejdet en god opgave sammen. Vi startede med at bygge vores frontend for at få siderne på plads i vores projekt, da vi allerede havde fået databasen i stand. Efter vi havde arbejdet med frontend og fået et semi færdigt produkt, gav vi backend alt vores fokus, hvor vi løbende prøvede at få frontend og backend til at interagere med hinanden, for at få en funktionel webapplikation. Vi havde nogle udfordringer hen ad vejen, samt vi fik ikke løst hele projektet, men vi gjorde vores bedste.

# Baggrund

På Bornholm har Olsker Cupcake gjort sig kendt for deres udsøgte udvalg af cupcakes, hvor du selv kan sammensætte din favorit kombination af bund og topping. For at gøre det endnu nemmere for deres kunder ønsker Olsker Cupcake en hjemmeside, hvor folk kan bestille deres personlige cupcakes og betale online.

Udover dette har de behov for en administrationsside, hvor de kan holde styr på kunder og ordrer, så deres forretning kan køre gnidningsløst. Kunderne vil kunne oprette en konto eller logge ind på hjemmesiden, vælge mellem et udvalg af forskellige cupcakes bunde og toppe med diverse smagsvarianter og kombinere dem efter deres præferencer. Efter bestilling og betaling kan kunderne derefter afhente deres skønne kreationer.

# Teknologi valg

## IntelliJ

- Intellij 2023.2.2
- Intellij 2023.3.3

## Java

- correto-17

Java 17 and correto-17 are compatible.

19.0.2

## PostgreSQL

- 42.7.2

## HTML

- HTML5

## CSS

- CSS3

## JDBC

- 4.2 API

## Thymeleaf

- 3.1.2.RELEASE

## Figma

- 116.17.12

## Javalin

- 6.1.3

# Krav

Firmaets håb med dette system, er at gøre det lettere for kunder at bestille deres yndlings cupcakes online på en hurtig og nem måde. De håber, at dette vil gøre kunderne gladere og samtidig styrke deres forretning ved at tilbyde en mere praktisk måde at bestille og betale for cupcakes. De vil også gerne have en del af systemet, der hjælper dem med at holde styr på ordrer og kundeinformation, så alt bliver mere organiseret internt. På den måde ønsker firmaet at skabe en bedre oplevelse både for kunder og medarbejdere, hvilket forhåbentlig vil føre til mere succes for deres bageri.

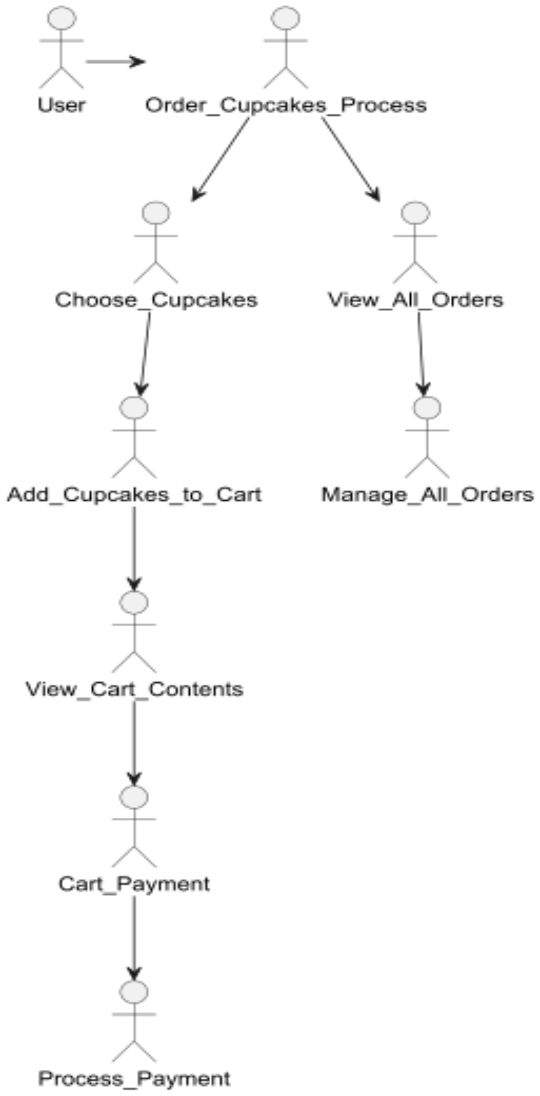
## User-stories

Alle user-stories er blevet udleveret i opgavebeskrivelsen, vi har ikke selv lavet dem samt haft kontakt med firmaet.

- **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 Postgres, så en kunde kan betale for sine ordrer.
- **US-4:** Som kunde kan jeg se mine valgte ordrelinjer 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 ordrelinje 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

Vi lavede et aktivitetsdiagram, da vi var færdige med vores projekt, for at beskrive overordnet hvordan vores forretningsproces virker over vores cupcake webapplikation.




Figur 1: Aktivitetsdiagram

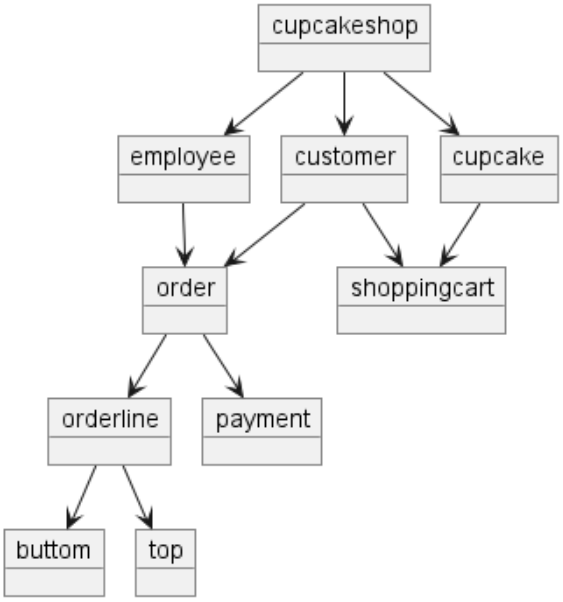
# Domæne model og ER diagram

## Domæne model

Denne domænemodel er lavet helt fra starten.

- Customer og cupcake er forbundet til shoppingcart for at repræsentere, at en customer kan tilføje cupcakes til indkøbskurven.
- Customer er bundet til order på grund af at en customer kan oprette en ordre.
- Employee er forbundet til order af den årsag, at en employee kan holde øje med customer orders.
- Order er forbundet til payment for at repræsentere, at en ordre skal betales.
- Order er også forbundet til orderline fordi, at en ordre kan have flere ordrelinjer.
- Hver orderline er forbundet til button og top, hvilket betyder at hver ordrelinje skal indeholde en bund og en top til cupcakes.

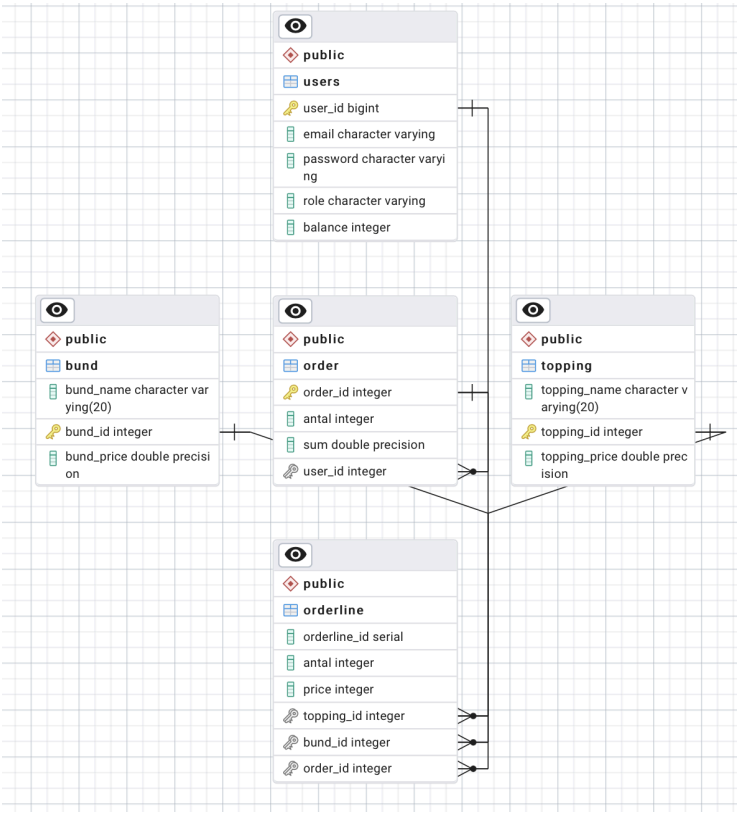
Det vi overvejede mest var hvordan relationen mellem order, og henholdsvis employee og customer skulle laves, da det er to forskellige user roles med forskellige adgange til systemet. Employee har adgang til nogle funktioner som customer ikke har, og customers funktioner er ikke så relevante for employee.




Figur 2: Domænemodel

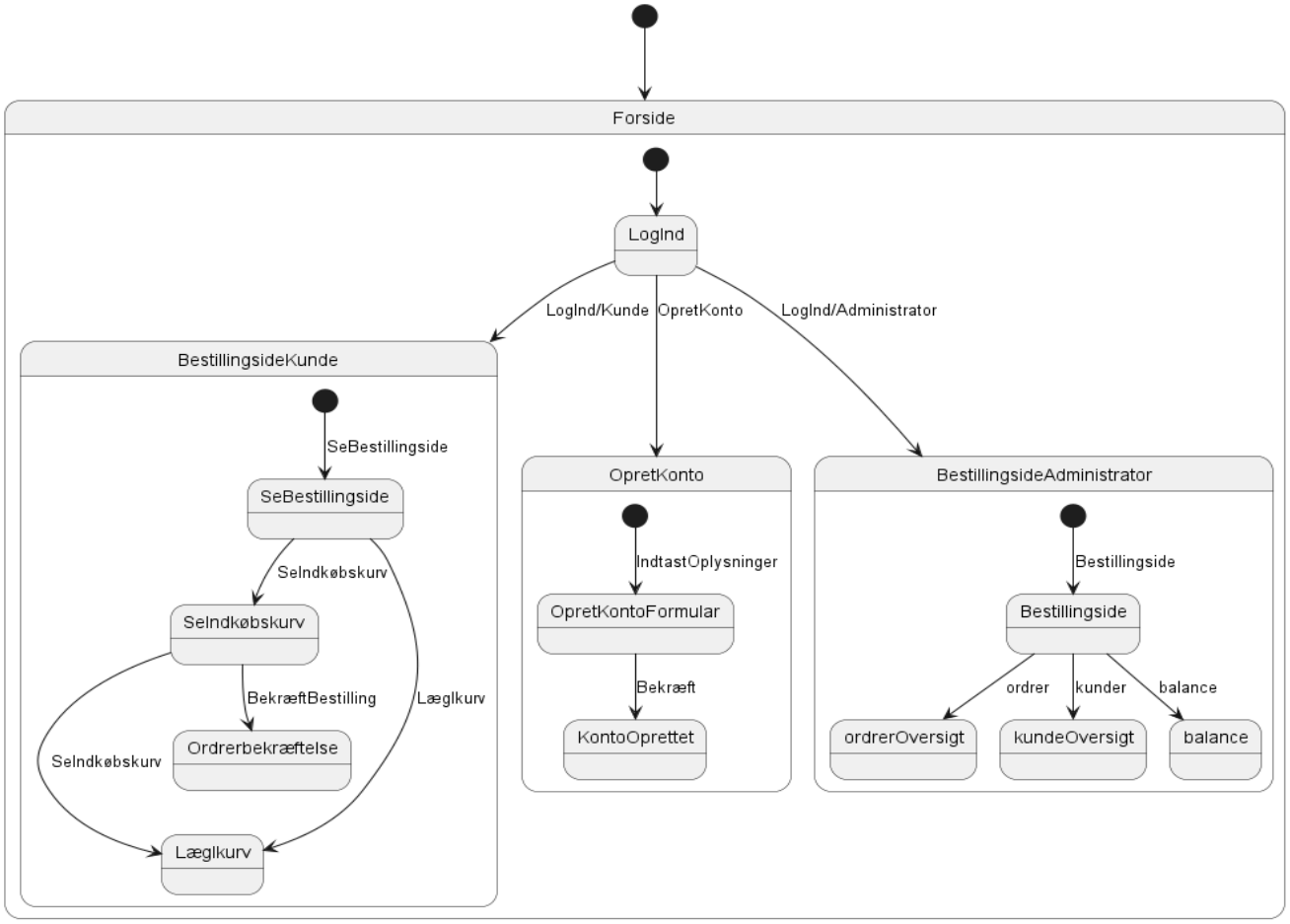
# ER diagram

Alle tabeller er i tredje normalform. De har primærnøgler og attributter. Hver ordrelinje er knyttet til en specifik ordre, og ved at bruge fremmede nøgler kan vi sikre, at vores data bevarer korrektheden og sammenhængen, og at hver linje kun tilhører én ordre. Vi undgår gentagne data i "Order" tabellen, ved kun at gemme detaljer om hver ordrelinje én gang i "OrderLine" tabellen. Det gør det også nemmere at forstå og vedligeholde tabellerne i databasen. Vi har lavet automatisk genereret ID for alle tabeller undtagen for bund og top, der er generelt ikke nogen grund, vi kunne også sagtens have tilføjet automatisk genereret ID så det er ikke med vilje, ville bare nævne det. Vi har lavet forbindelserne for at tillade flere veje at nå fra et sted til et andet, for eksempel kan man gå fra en user til deres orderlines via deres orders, dette giver en mere fleksibel måde at få adgang til og manipulere data, dog øger det også kompleksiteten af databasen.




Figur 3: ER diagram

# Navigationsdiagram




Figur 4: Navigationsdiagram

# Særlige forhold

Det som gemmes i sessionen er de valg brugerne taster ind i formularen når de bestiller cupcakes som top, bund, og antal. Den kunde som er logget ind gemmes også i sessionen, så data bliver knyttet til den pågældende kunde. ShoppingCartListen bliver også gemt med de forskellige data omkring bestillingen. I vores projekt håndterer vi exceptions ved brug af DatabaseException når der hentes data fra databasen og når der sættes data ind i databasen, samt vi brugte exceptions til at håndtere validering af brugerinput.

# Status på implementation

Vi har ikke nået at blive færdige, vi havde nogle mangler samt fejl, den fejl vi havde fundet i sidste øjeblik var at vi ikke fik givet administrator tilladelser til at tilgå vores 3 knapper, Order, Kunder og Balance som kun bør være administratorer som kan tilgå og se dem. Vi fik ikke nået at færdiggøre det med at når man trykker på Confirm order knappen at det så skulle blive sendt til databasen så vi fik ikke fuldt færdiggjort vores backend kun frontend.

# Proces

Vi havde en plan om at få styr på, hvad vi skulle lave og hvordan vi skulle gøre det. Vi ville også sikre os, at vi alle var enige om, hvad vores forskellige opgaver var, og hvad vi skulle opnå.

Undervejs i projektet lykkedes det os at blive enige og få det hele til at fungere. Vi gjorde vores bedste, og det var nemt at snakke sammen og samarbejde. Vi støttede hinanden, hvilket hjalp os med at løse problemer.

Det gik godt, at vi kunne kommunikere og samarbejde effektivt. Støtten til hinanden var også positiv. Dog kunne vi have gjort det bedre fra start. Det ville have været smart at starte projektet mere struktureret og sørge for, at vores kode var mere ensartet fra begyndelsen. Det ville have undgået en del forvirring, især med navngivningen af tingene.

En af de ændringer vi vil foretage os næste gang, er at få snakket tingene igennem fra start, så der ikke kommer forvirringer, for eksempel angående database, mest for at alle ved præcist hvordan vores projekt skal foregå.

# Video demo

Youtube link til video demo: https://youtu.be/ylgzYTH-aC0