

# **CUPCAKE FORSIDE 2. SEMESTER F23**

## **DELTAGERE:**

**Salar Salim**

**Email:** cph-ss647@cphbusiness.dk

**GitHub:** Salar86

**Michael Meyer**

**Email:** Cph-mm667@cphbusiness.dk

**GitHub:** havvand

**Rasmus Jensen**

**Email:** cph-rj205@cphbusiness.dk

**GitHub:** rasm221f

**Hold og gruppenummer:** Hold A, gruppe 6.

**28/3/2023**

# Indledning

Vi har fået til opgave, at lave et hjemmeside-design til Olsker Cupcakes, hvis funktionalitet skal kunne imødekomme en række krav, stillet af udbyderen. På hjemmesiden skal man kunne logge ind, enten som administrator eller bruger, og afhængig af hvilken man er logget ind som, blive mødt med forskellige muligheder.

Derefter skal man som kunde kunne sammensætte cupcakes, som man lyster, med vilkårlige toppe og bunde, og kunne se og redigere i sin indkøbskurv.

Som administrator skal man kunne se alle kunder, indsætte penge på deres konto, og have en oversigt over alle ordrer.

## Baggrund

### Kunde:

Olsker Cupcakes er en bagervirksomhed fra Bornholm med fokus på Cupcakes bagt med økologiske ingredienser.

### Kundes krav:

Kunden ønsker en hjemmeside med en række forskellige funktionaliteter. Herunder:

1. Online bestilling af bruger-sammensatte cupcakes.
2. Oprettelse af bruger-konti med historik, samt betaling.
3. Ejeren af løsningen skal igennem løsningen have overblik over alle ordrer. Vedkommende skal også kunne slette forfaldne ordrer, der for eksempel aldrig er blevet betalt for.
4. Ejeren af løsningen skal have direkte adgang til SQL-databasen, så denne kan foretage nødvendige ændringer i samme.
5. Som kunde skal man kunne se sin indkøbskurv, inden man betaler, så man kan se den samlede pris på alle varer man har lagt i. I forlængelse af dette skal man også kunne redigere i sin indkøbskurv, hvis man ønsker at fjerne en vare man har lagt i.
6. Som bruger, både som administrator, og kunde, skal man kunne se sit brugernavn displayet et sted på hjemmesiden.

# Teknologivalg

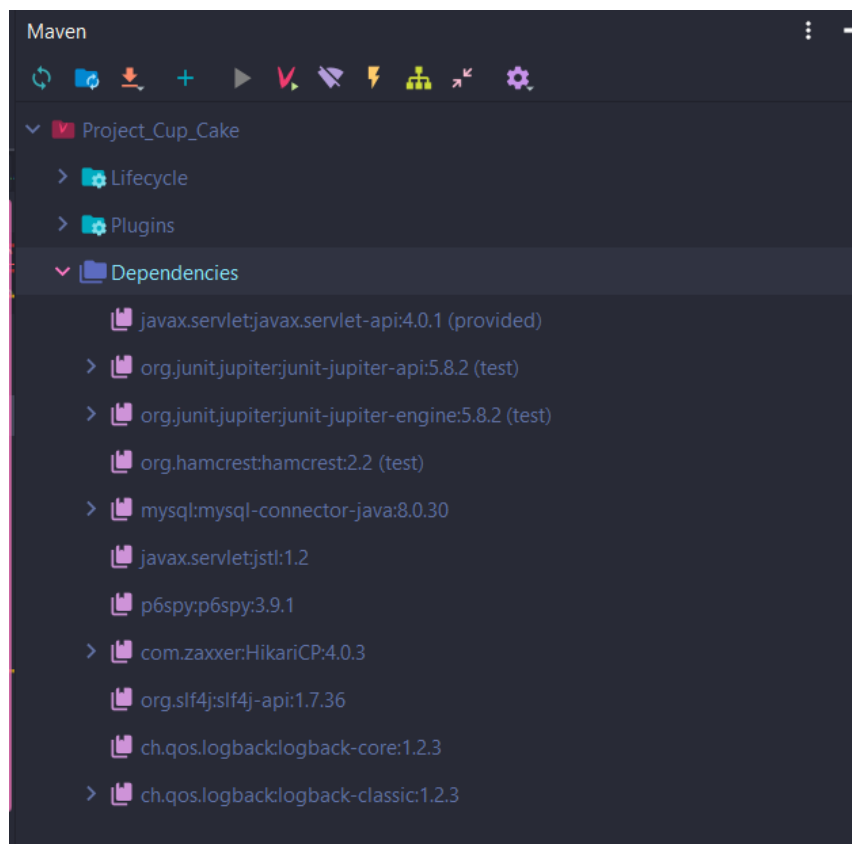
IntelliJ IDEA 2021.2.4 Ultimate Edition

- Build #IU-212.5712.43

MySQL

- MySQL Workbench 8.0 CE

Maven



Tomcat

- 9.x

MySQL Connector JAVA

- V. 5.1.49.

Java version:

- Vi har testet med java version 8 og 11, men projektet kan formentlig også køres med nyere versioner.

# Krav

## **Firmaets håb:**

Firmaet håber, med vores hjælp, at have en fungerende hjemmeside, der kan realisere deres iværksætterdrømme. Vores løsning skal kunne gøre, at de som iværksættere blot skal bekymre sig om deres bagværk, og sørge for det administrative, i forhold til at holde styr på kunder i systemet og følge op på ordrer.

## **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 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).

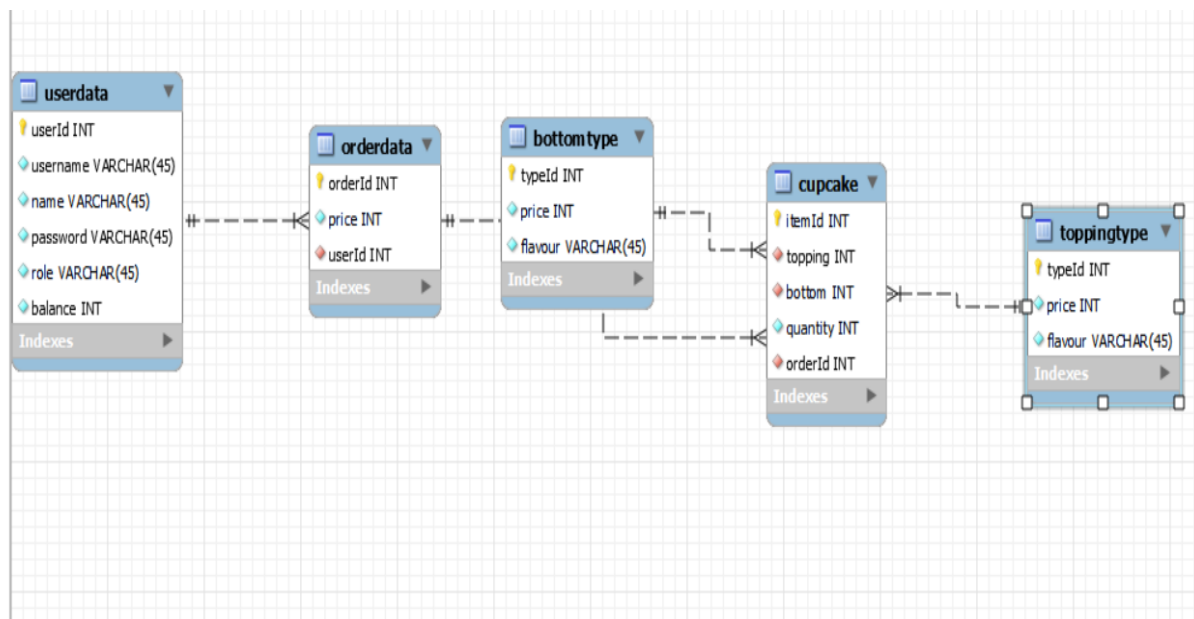
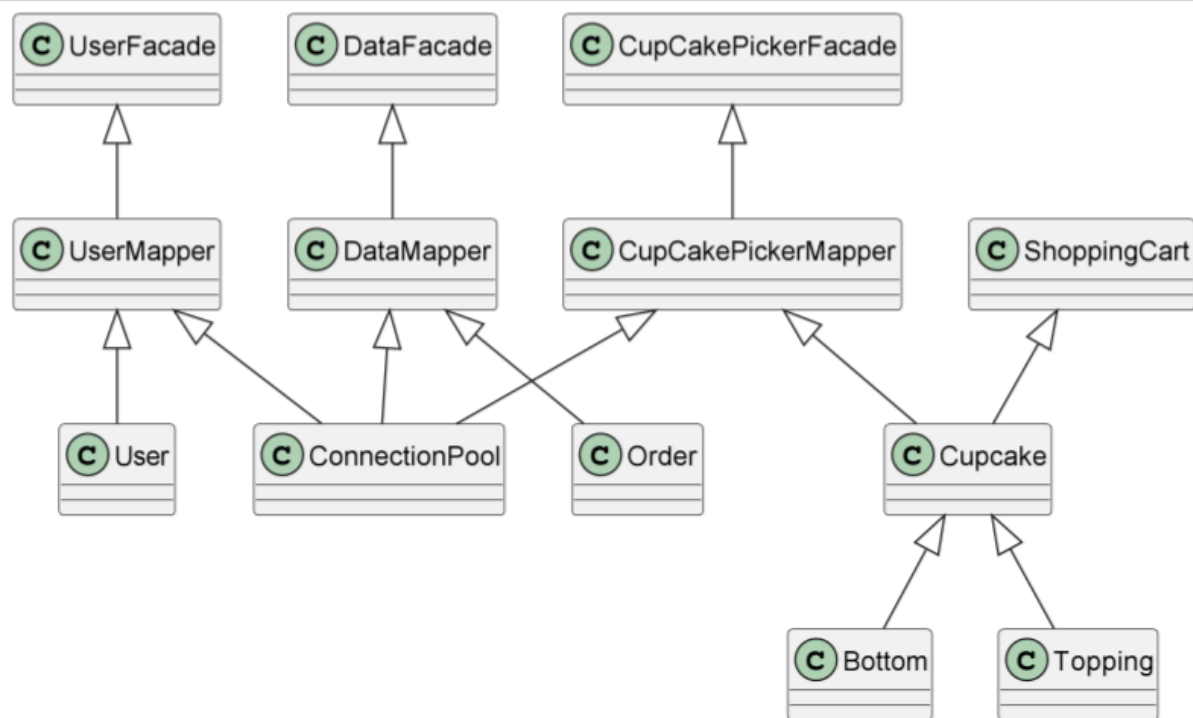
US-6: Som administrator kan jeg se alle ordrer i systemet, så jeg kan se hvad der er blevet bestilt.

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.

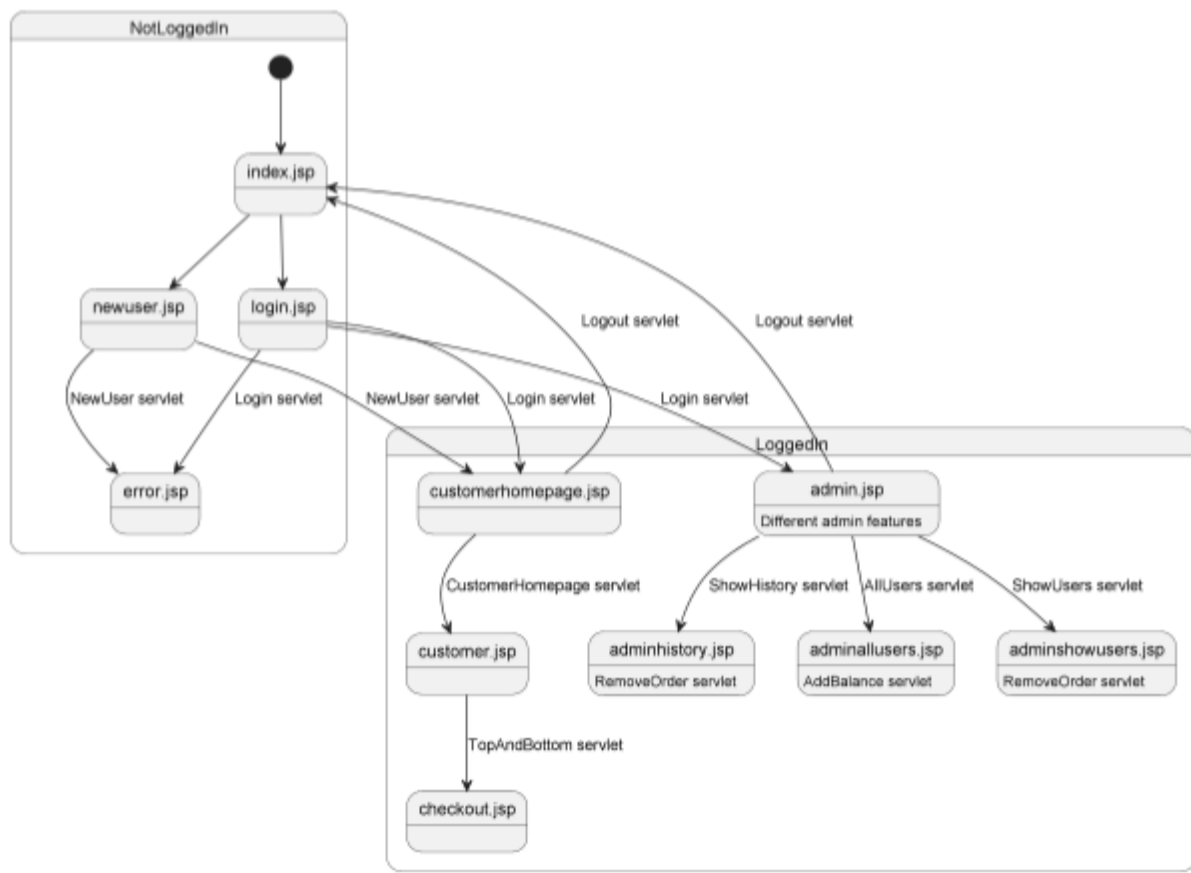
US-8: Som kunde kan jeg fjerne en ordre fra min indkøbskurv, så jeg kan justere min ordre.

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

# Domæne model og ER diagram



# Navigationsdiagram



## Særlige forhold

Der er lavet en særklasse, der hedder "DatabaseExceptions", i dens egen package "Exceptions", som vi kan instantiere, og kalde efter brug. Den extender "Exceptions", og udover dette logger den fejlmeddelelsen.

Validering af brugerinput kan ses for eksempel i vores "New User" servlet, hvor vi har gjort at, for at få lov til at lave en bruger skal man skrive sit password ind to gange, og kun hvis disse passwords er identiske, bliver brugeren oprettet. I forlængelse heraf, kunne vi have benyttet os af String.length metoden, og sørget for, at passwordet skulle have en minimumslængde på x antal karakterer. Vi kunne også have brugt String.Contains metoden, og sørget for, at passwordet skal indeholde for eksempel specielle karakterer eller tal, eller begge.

Vi kunne også have brugt secureString, i stedet for String, til vores passwords, så de blev krypteret.

Desuden er alle vores databasekald PreparedStatements, så vi er sikret mod SQL-injections, så hackere ikke kan tilgå data, der ikke er tilgængeligt for dem.

Alle de sider af vores website, vi ikke ønsker brugere, der ikke er logget ind at kunne tilgå, har vi lagt ind i vores web-inf folder, så det kun er autoriserede brugere, der har adgang til disse.

## Status på implementation

Vi fandt en fejl på sidste dagen, der gjorde det umuligt at bestille en cupcake med både chokolade-topping og chokolade-bottom. De var ikke tilknyttet samme variabel eller værdi, vi prøvede at lægge dem i hver deres div klasse, de blev itereret i to forskellige for each loops, men intet gjorde det muligt. Vi prøvede tilmed at lave en cupcake entitet, og først samle cupcaken der, men det løste stadig ikke problemet.

Til sidst ændrede vi navnet på bunden i databasen, så den ikke længere hed chocolate, men nu hed chocolate cake. Dette løste det.

Vi overvejede også at tilføje “custom” buttons på admin siden, så man kunne navigere rundt ved hjælp af knapperne oppe i højre hjørne, i stedet for at have dem midt på side, men da vi prøvede dette, blev vi sendt videre med den rigtige servlet, men displayet var blankt. Vi valgte derfor at gå med det UI, der gjorde at admins kun kunne navigere ved hjælp af knapperne i midten.

## Proces

Vi har i forvejen arbejdet sammen som gruppe på tidligere semester, og vi har derfor allerede en struktur i forhold til mødeform. Vi mødes fysisk, hver dag klokken 9, for bedst at kunne sparre, og søge hjælp hvis nødvendigt.

Til at starte med diskuterede vi, hvordan vi skulle gøre i forhold til samarbejde, og prøvede at lægge en plan i forhold til hvordan programmet skulle se ud, hvordan man skulle navigere, og til at sætte et fælles GitHub repository op, så vi kunne arbejde individuelt.

Dernæst uddelegerede vi opgaver, og forsøgte at dele opgaverne op imellem os, i henhold til interesser og styrker.

Dette gik udmærket, dog med visse huller i vejen, såsom sygdom og arbejde, der komplicerede dette.

Vi ville ønske, at vi kunne have brugt dagen, hvor vi fik udleveret opgaven, til at gennemgå nogle af elementerne vi ikke havde stødt ind i før, og gennemgået disse lidt mere i dybden. For eksempel “pagetemplate.tag” filen, der gør grundlaget for brugergrænsefladen.

Årsagen bag, og “good practice” i forhold til hvad en “config” package, hvad en “control” package og hvad en “persistence” package bør indeholde mm.

Forklaring på en web-inf folder, og hvad der være i denne, og hvad der ikke behøves.

## Videolink til userstories:

<https://youtu.be/CpfHrhVAvog>