

# Olsker Cupcakes



## Indledning og Baggrund

I dette projekt har vi fået til opgave at stable en hjemmeside på benene for et par iværksættere fra København, herefter refereret til som IFK. De ønsker at starte op en cupcake forretning på Bornholm. Ideen bag forretningen er at en kunde skal kunne bestille cupcakes online, derefter hente dem i jeres butik i Olsker.

IFK ønsker at deres kunder via deres hjemmeside skal kunne bestille cupcakes med valgfri top og bund. For at kunne bestille skal kunden oprette en bruger med email og password på hjemmesiden. Kunden benytter seg af e-mail og password for at logge seg på hver gang han/hun besøker siden.

I stedet for at betale med bankkort for hver bestilling kunden foretar sig, ønsker IFK at kunden fyller på sin konto med et beløp de kan handle for. Kunden vil ha tilgang til å se alle sine ordre, her vil kunden også kunne se hvilke ordre som er gjennomført.

IFK ønsker også at en administrator skal ha tilgang til websiden. Når administrator logger på vil han/hun komme inn på en egen administratorside. På denne siden skal administratoren kunne se alle kunder i systemet og holde styr med de ordre der bliver foretatt av de ulike kunder. Administratoren skal kunne gi tilbakemelding til kunden på status på en ordre. Administratoren skal også kunne indsætte et beløp på en kundes konto.

## Teknologi valg

- IntelliJ 2019.3.3
- Java 1.8
- JDBC
- MySql
- Maven
- Tomcat 8.5.50

## Krav

*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 e-mail og kodeord. Når jeg er logget på, skal jeg kunne min email på hver side.

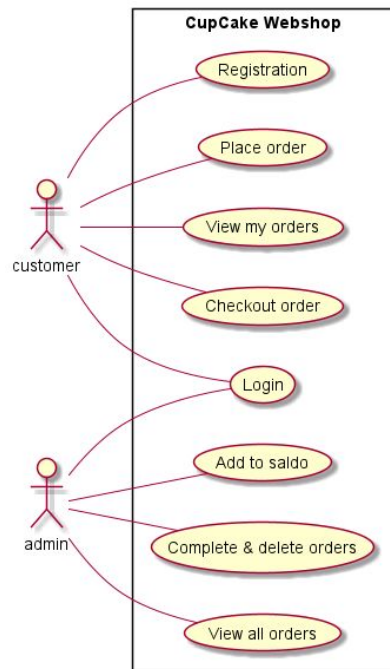
*US-6:* Som administrator kan jeg se alle ordrer i systemet, så jeg kan se hvad der er blevet bestilt.

*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.

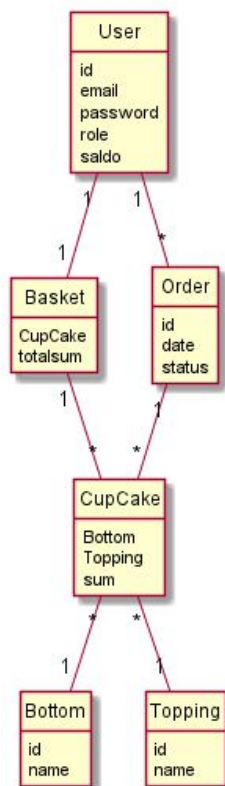
*US-8:* Som kunde kan jeg fjerne en ordre fra min indkøbskurv, så jeg kan justere min ordre.

*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.

Use case:



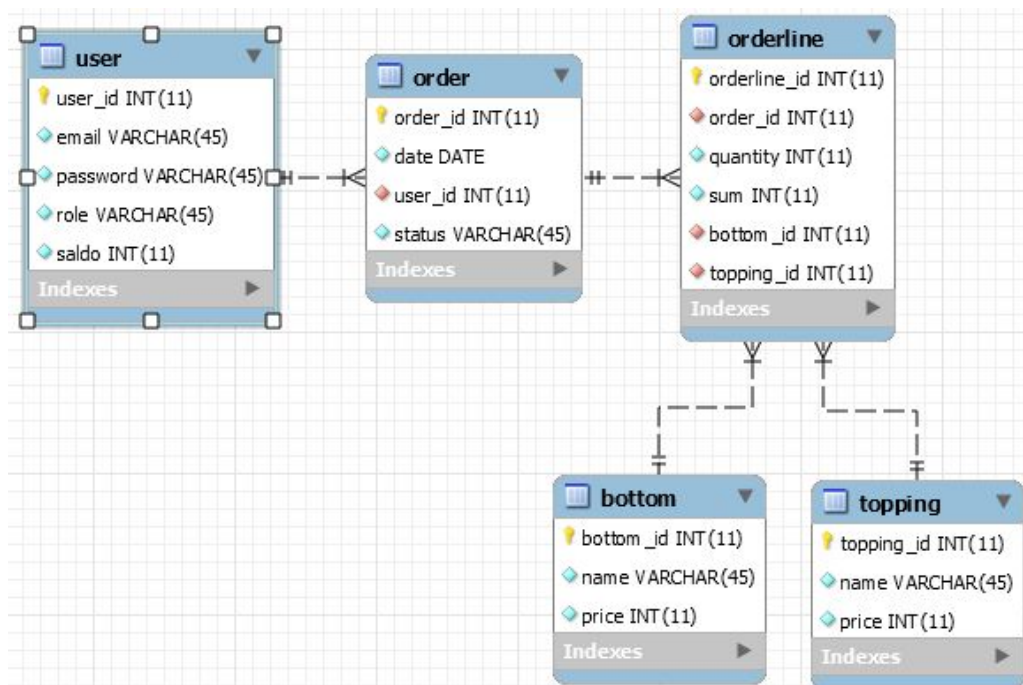
## Diagrammer



### Domæne:

User kan lave en ordre, der indeholder en eller flere cupcakes, som hver især består af én bottom og én topping. Disse ordrer placeres i en kurv. En user har kun en kurv, og forholdet er derfor 1 til 1. Alle ID er autogenereret.

EER:



User:

Useren har et autogenerated ID, som bliver genereret, når en ny user bliver oprettet. Email og password er selv valgt og knyttet til deres konto. Rollen på almindelige kunder er "Customer" og der er en admin role. Saldo indeholder det beløb som kunden kan handle for.

Order:

Order ID bliver autogenerated ved ny ordre. Date bliver sat til bestillingsdatoen. Ordren bliver knyttet til kunden ved hjælp af en foreign key på user ID'et. Status viser statussen på bestillingen, d.v.s. om den blot er bestilt eller afsluttet.

Orderline:

Vores kurv består af 1 eller flere orderlines. Orderline ID'et er autogenerated og bliver tilknyttet Order ID via en foreign key. Desuden består en orderline af en topping, bottom og quantity - altså antal af valgte cupcakes - og en sum. Hver topping og bottom har et unikt ID, der linker dem sammen med orderline vha. foreign key.

## Navigation:

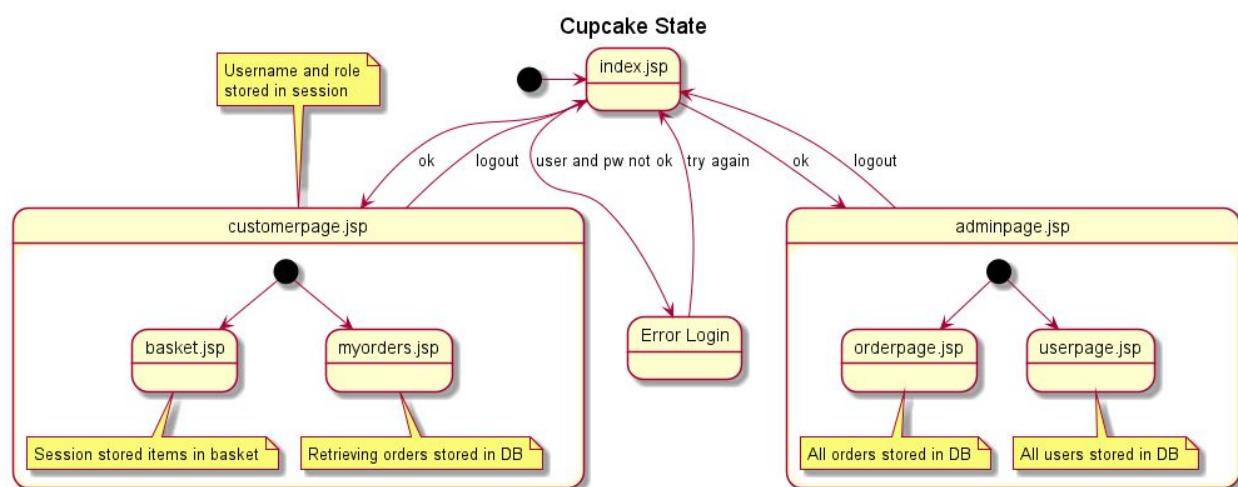
Fra index.jsp, som fungerer som loginsiden, kan man enten logge på som kunde eller admin. Hvis legitimationsoplysningerne er forkerte, mødes man med en fejl og sendes tilbage til index.jsp. Login gemmes i sessionscope.

Som customer får man efter login adgang til at bestille cupcakes på customer.jsp.

Derfra kan man vælge at se sin kurv (basket.jsp), som gemmes i sessionscope.

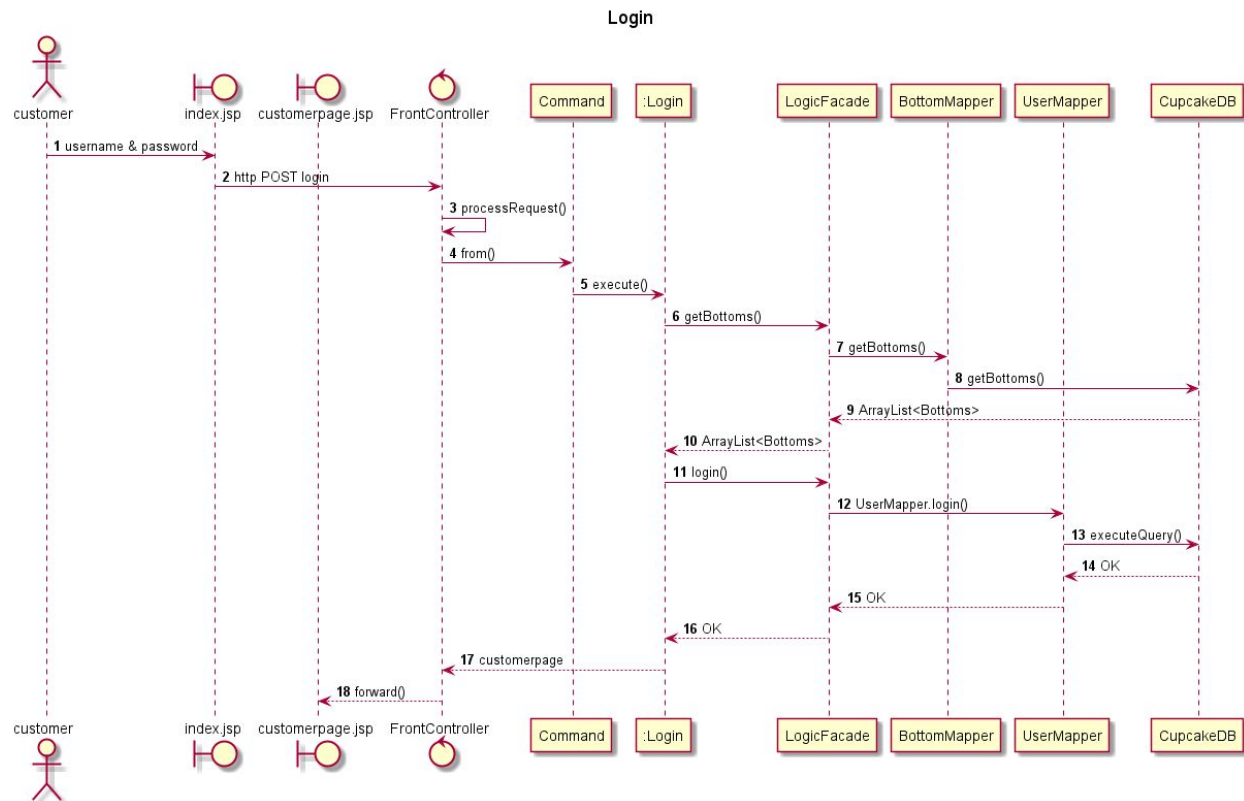
Derudover kan man se tidligere ordre (myorders.jsp), som hentes ned fra databasen. I basket.jsp foretager man købet. Logout sender en tilbage til indexsiden.

Som admin kan man se en oversigt over alle users (userpage.jsp) samt alle kundeordrer (orderpage.jsp), der begge hentes ned fra databasen.



## Sekvens - Login:

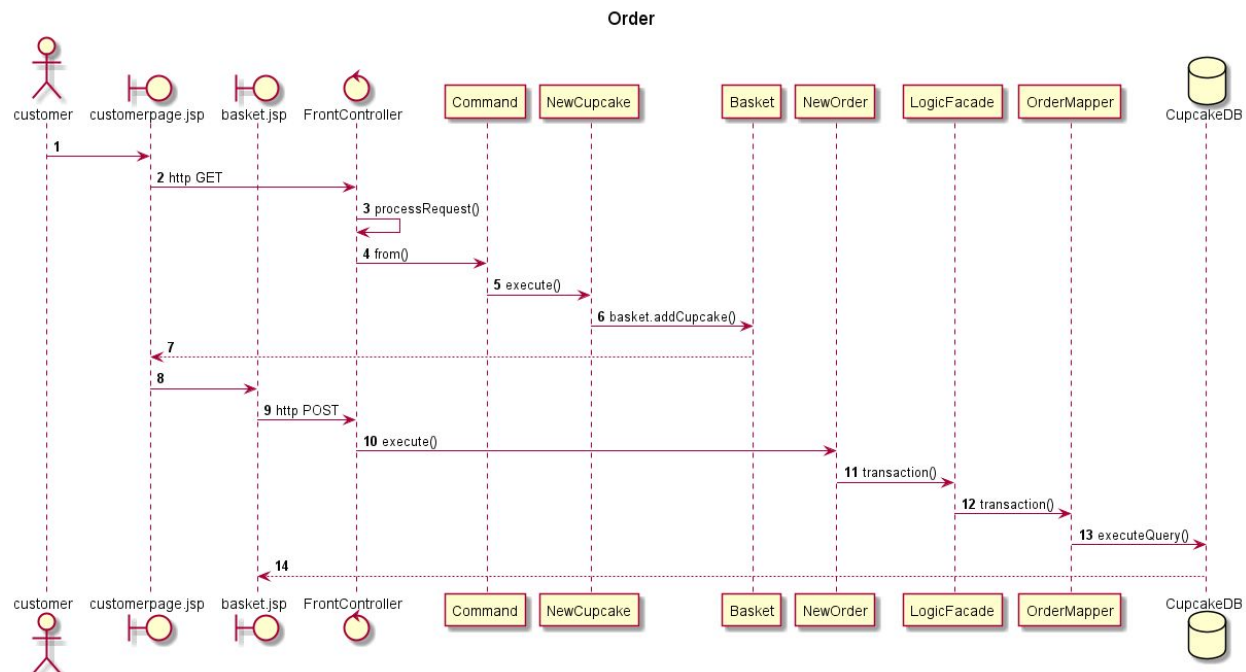
Input fra customer (username & password) sendes fra index.jsp videre til FrontController via en http POST, behandles af metoden `processRequest()` og derefter kaldes Command-interfacet med `from()`-metoden, hvorefter login udføres vha. `execute()`. `Login()` kalder `LogicFacade`, som kalder `UserMapper.login()`, og når ned til DB vha. `executeQuery()`. Hvis email og kodeord findes i DB, sendes man til **customerpage.jsp** igennem FrontController, der benytter sig af `forward()`. Samtidig med login, hentes bottom- og toppingarraylister fra DB igennem `LogicFacade`, der indeholder metodekaldene `getBottoms()` og `getTopping()` og lægges i `applicationscope`, hvis de ikke allerede findes der.



### Sekvens - Order:

Når en kunde laver en ny cupcake sendes den videre til FrontController via http GET, behandles af metoden `processRequest()`, `Command`-interfacet kaldes med `from()`-metoden, og `NewCupcake` udføres af `execute()`. Den tilføjes til Basket med `basket.addCupcake()` og sendes derfra tilbage til `customerpage.jsp` og ender så i `basket.jsp`.

Så snart ordren er købt, sendes data videre vha. http POST, hvor `execute()` fyres af. Fra `NewOrder`, der går indenom `LogicFacade`, kaldes metoden `transaction()`, der får indsat ordre og orderlines i DB via `executeQuery()`.



## Særlige forhold

Alle user-informationer og basket ligger i session. Bottom- og toppinglisterne ligger i application.

Validation af brugerinput er håndteret ved at gøre brug af dropdown-menuer, når en cupcake kreeres.

## Status på implementation

Vi mangler en .jsp-kvittering side.

Derudover mangler vi en del styling, især på orderpage.jsp mht. knappernes placering.