



## Hold Medlemmer Gruppe 9

17 November 2022

|                         |                           |   |
|-------------------------|---------------------------|---|
| Mayas Fares             | cph-mf337@cphbusiness.dk  | <a href="https://github.com/mayasfares1/">https://github.com/mayasfares1/</a> |
| Mohamed Faqi Salim      | cph-ms1068@cphbusiness.dk | <a href="https://github.com/MrSalim86">https://github.com/MrSalim86</a>       |
| William Emil Yus Hansen | cph-wh1113@cphbusiness.dk | <a href="https://github.com/asphixia">https://github.com/asphixia</a>         |

## Table of Contents

|                          |   |
|--------------------------|---|
| Indledning .....         | 3 |
| Teknologivalg .....      | 3 |
| Krav .....               | 4 |
| Aktivitetsdiagram .....  | 5 |
| EER Diagram .....        | 6 |
| Navigationsdiagram ..... | 7 |
| Proces .....             | 7 |
| Særlig forhold .....     | 8 |

## Indledning

Hovedformålet for Cupcake projektet og rapporten er at udvikle en simpel webshop ud for nogle krav fra kunden. Olsker cupcakes kunder skal kunne bruge hjemmesiden til at købe og bestille cupcakes som kan hentes efterfølgende på butikken.

## Teknologivalg

- IntelliJ 2021.2.4
- MySQL Workbench 8.0CE
- Apache TomCat Server 9.0.67
- Figma 9.0
- Java 11
- Maven
- HTML5
- CSS3
- Git/Github

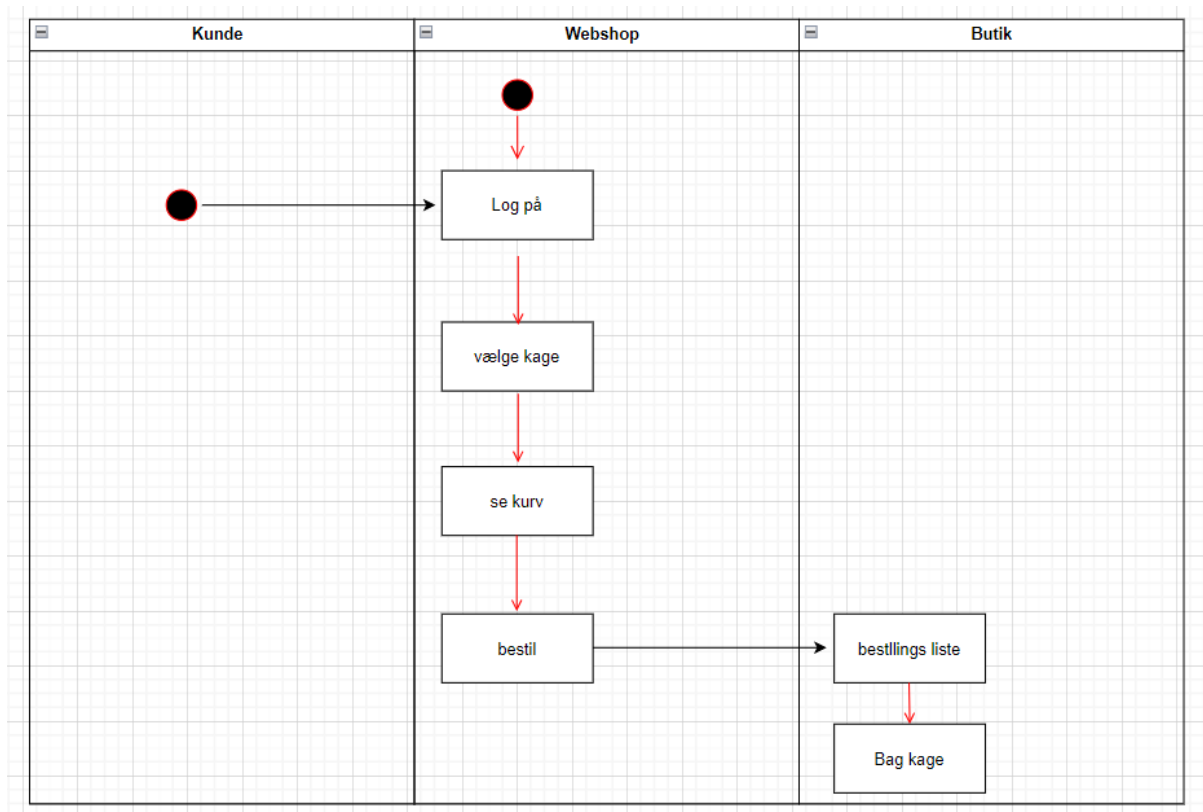
Vi har begyndt vores projekt med at lave en hurtig mockup på tavlen og efter vi blev enige om strukturen så startede vi med at tegne vores mockup i Figma. Selve designet af databasen var lavet i MySQL workbench. Vi begyndt med samme procedure som før. Vi tegnede ERR diagrammet på tavlen og derefter ind i MySQL workbench. Selve programmeringsdelen var udført i Intelli J med TomCat Server som skaber forbindelsen.

## Krav

Kunden har planer til at udvide sit kundesegment. En af det bedste måder at gøre det på er med en Webshop. På den måde så kan kunden får kontakt med flere potentielle kunder i et breder geografisk område. For at kunne tilfredsdele kundens behov så er der en del **usecases** som skal løses.

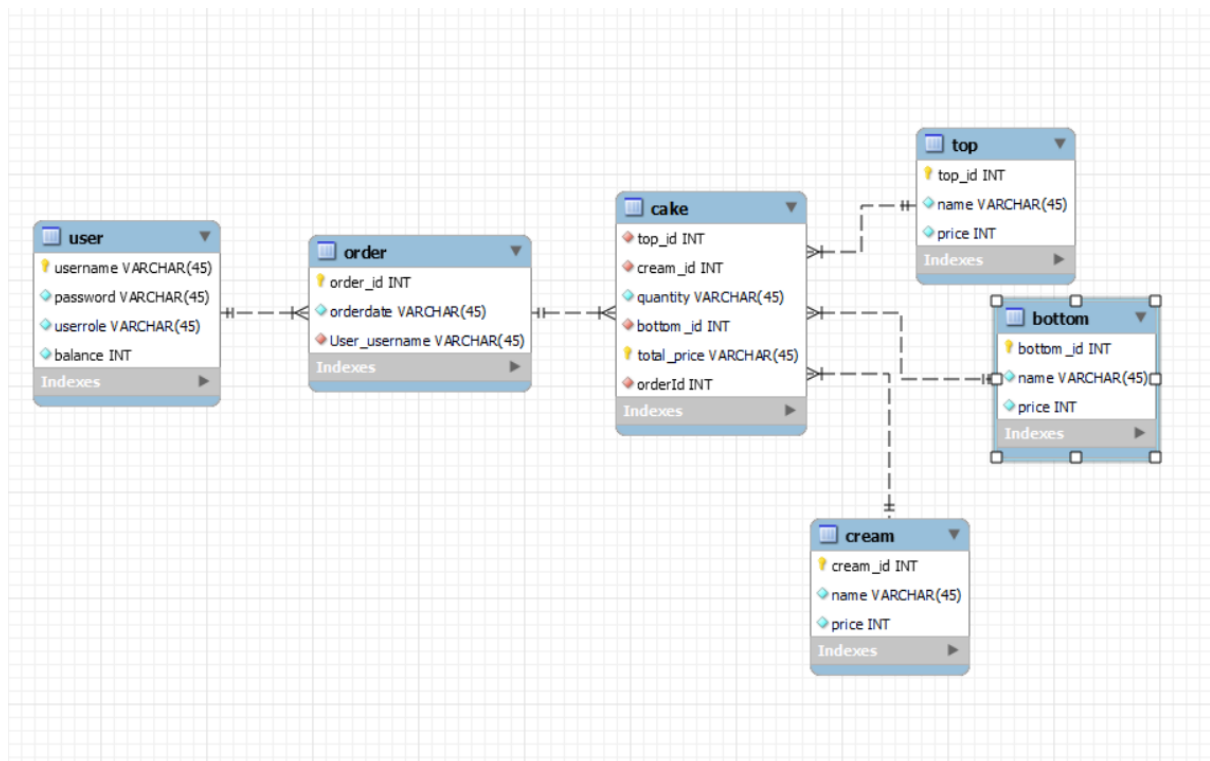
- US-1: Som kunde kan jeg bestille og betale cupcakes med en valgfri bund og top, sådan at jeg senere kan køre forbi butikken i Olsker og hente min ordre.
- US-2 Som kunde kan jeg oprette en konto/profil for at kunne betale og gemme en ordre.
- US-3: Som administrator kan jeg indsætte beløb på en kundes 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 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.

## Aktivitetsdiagram



Kunden logger ind på hjemmesiden med et username og adgangskode, efter dette så forsætter kunden med at bestille sine kage/kager med forskellige bund og toppings. Kunden kan efter valg af kage se hvor meget der blev valgt. Efter kunden trykker på bestil knappen så ryger bestillingslisten over til butikken hvor butikken kan gå i gang med at bage.

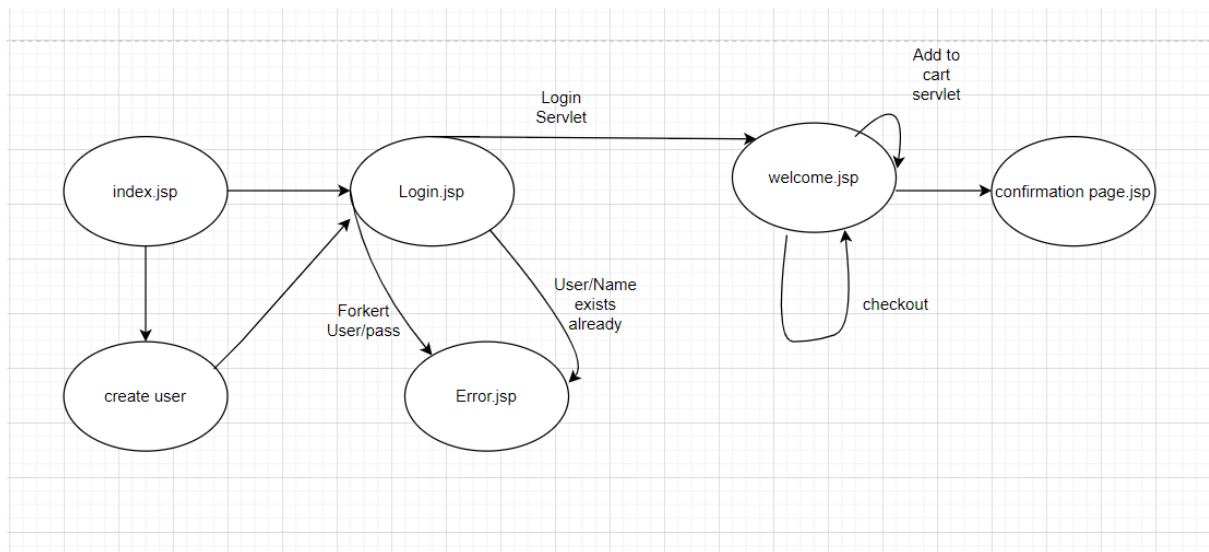
## EER Diagram



Vores EER-diagram starter med **user** hvor *username* er vores primary key som skaber en forbindelse til *username* (i **order**) som er vores *foreign key*. **User** har attributter som *username*, *password*, *userrole* og *balance*. Der bliver skabt en forbindelse fra **Order** vha. *order\_id* som er primary key i **Order** den skaber en forbindelse til *orderId* (foreign key) i **cake**. **Order** har attributter som *order\_id*, *orderdate*, *username*. **Cake** skaber det samme forbindelser til både **top**, **bottom** og **cream**.

Vi har desværre ikke haft mulighed for at indrette vores database efter 3. normalform pga. tidsbegrænsninger for aflevering af opgaven.

## Navigationsdiagram



Kunden starter på `index.jsp` siden. Kunden bliver bedt om at *login* eller *create user*. Hvis kunden skriver forkert, så kastes der en exception. Derefter så forsætter kunden videre til `welcome` (vha en `Login` servlet) siden hvor valgmulighederne bliver videreført til kurven vha. (`add to cart` servlet). Efter dette så går kunden videre til `confirmation page` hvor kvitteringen udskrives.

## Proces

Vi har begyndt på projektet med at lave en hurtig skitse over hvordan webshoppen skal se ud. Vi tegnet et mockup som viser nogenlunde hvordan hjemmesiden skal se ud. Efter vi har aftalt strukturen så begyndt vi med at diskutere hvordan databasen repræsenteres. Vi har igen lavet en mockup hvor vi fik den grundlæggende struktur på plads. Derefter begyndt vi med at indtaste relationer i MySQL. Det var der vi fik udfordringer med diverse SQL fejl. Vi har prøvet at køre parallelt med uddeling af opgaver, men det var begrænset hvor meget man kan lave når databasen var ikke funktionelt.

## Særlig forhold

- Information fra user bliver gemt i session
- Vi håndterer exceptions vha. intelli j try/catch handler
- Vi har brugt *UserException* funktionalitet for at håndtere forkerte indtastninger, brugen får en fejlbesked.
- Vi har brugt *DatabaseException* til at håndtere fejl indtastninger.

## Status på implementation

- Vi har nået at lave Usercase stories fra 1-6
- Vi mangler at finpudse front end delen
- Mangler at finpudse vores figma model
- Mangler at lave tests på vores kode