

Cupcake 2024



Mads Benjamin Ribberholt - **cph-mr415@cphbusiness.dk** - MadsBenjaminR

Daniel Herlev Moldavsky - **cph-dm189@cphbusiness.dk** - moldamolda

Jeppe Rønning Koch - **cph-jk469@cphbusiness.dk** - JeppeKoch

Laith Alkaseb - **cph-la356@cphbusiness.dk** - MINGO-INC

Klasse A

2/4 -12/4 2024

Indholdsfortegnelse

Indledning.....	3
Baggrund	3
Kundens krav.....	3
Valg af teknologi	4
User Stories	4
Aktivitetsdiagram.....	5
Domænenemodell or ERD	5
Domænenemodell	5
ERD	6
Navigationsdiagram.....	7
Særlige forhold	8
Status på implementering.....	9
Proces	10
Link til Youtube	10

Indledning

Projektet har til formål at udgøre en implementering af en webside for Olsker Cupcakes ud fra givne krav og behov. Projektet sigter mod at skabe en brugervenlig og intuitiv hjemmeside, der gør det muligt for kunden at købe cupcakes og butiksejere at holde overblik over deres forretning. Målgruppen for rapporten vil være en fagfælle som ikke kender til opgaven, og rapporten har derfor til formål at introducere denne til projektet. Rapporten vil også præsentere de vigtigste krav og behov givet af Olsker Cupcakes.

Baggrund

Virksomheden er et dybde-økologisk iværksættereventyr fra Bornholm, der sælger cupcakes fra en fysisk butik. Konceptet er, at man som kunde valgfrit kan sammensætte en top og bund, der i kombination former en komplet cupcake.

Kunden har de seneste år oplevet en stigende efterspørgsel. En hjemmeside, hvor man kan bestille og købe cupcakes før afhentning, vil derfor møde efterspørgslen og lede til højere omsætning.

Kundens krav

- Kunde kan bestille og betale for cupcakes med valgfri bund og top
- Kunde kan oprette en konto eller profil på hjemmesiden for at betale for ordre
- Administrator kan tilføje beløb direkte til kundens konto for at muliggøre betaling af ordre
- Kunde kan se de valgte ordrelinjer i en indkøbskurv, samt den samlede pris for ordre
- Kunde og administrator kan logge ind i systemet ved hjælp af deres e-mail og adgangskode. Efter login skal kundes e-mail vises på alle sider
- Administrator kan se en liste over alle ordrer i systemet for at holde styr på ordrer

Valg af teknologi

- IntelliJ IDEA 2023.3.5
- Java 11.0.20
- HTML
- CSS
- Maven 17
- Jetty 11.0.20
- Javalin 6.1.3
- Javalin-rendering 6.1.3
- Thymeleaf 3.1.2.RELEASE
- Thymeleaf-extras 3.0.4.RELEASE
- PostgreSQL 42.7.2
- PgAdmin 4 version 8.3
- Hamcrest 2.2
- Kotlin 1.9.22
- Junit 5.10.2
- HikariCP 5.1.0
- Jackson 2.17.0-rc1
- Slf4j 2.0.12

Hosting

- Docker 4.27.2
- Linux VM 7.0.12

User Stories

Som tidligere nævnt har kunden de seneste år oplevet en stigende efterspørgsel på deres cupcakes. En hjemmeside, hvor man kan bestille og købe cupcakes før afhentning, vil derfor møde efterspørgslen og lede til højere omsætning hos virksomheden.

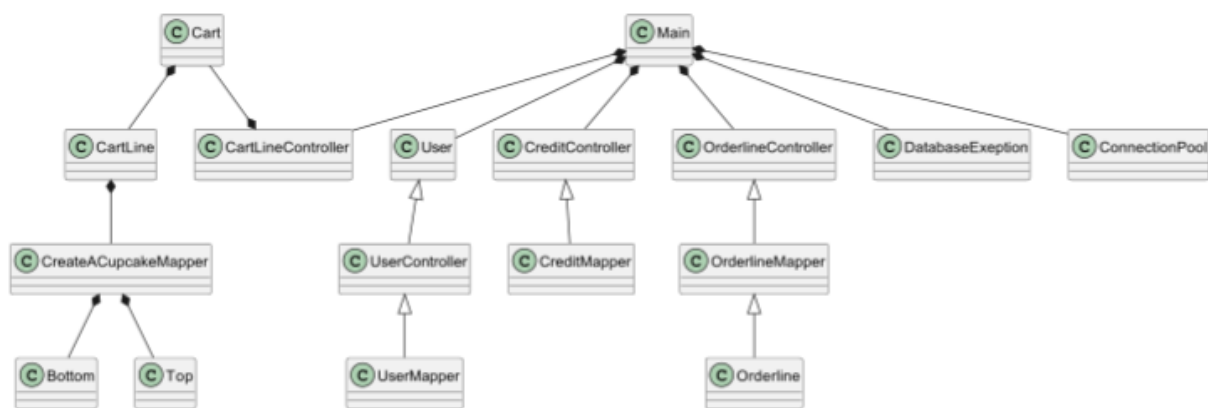
- US-1: Som kunde kan jeg bestille og betale cupcakes med en valgfri bund og top, sådan at jeg senere kan køre forbi butikken i Olsker og hente min ordre.
- US-2 Som kunde kan jeg oprette en konto/profil for at kunne betale og gemme en en ordre.
- US-3: Som administrator kan jeg indsætte beløb på en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

Det er blevet besluttet ikke at inkludere et aktivitetsdiagram. Vi vurderer at det er mest optimalt at lave i starten af projektet.

Domænemodel or ERD

Domænemodel



Fordi man som bruger ikke skal være logget ind for at sammensætte en cupcake og se din indkøbskurv, var der ikke brug for at lave en CartLinkMapper der skulle gemme noget i databasen, dette sker i OrderlineMapperen.

ERD

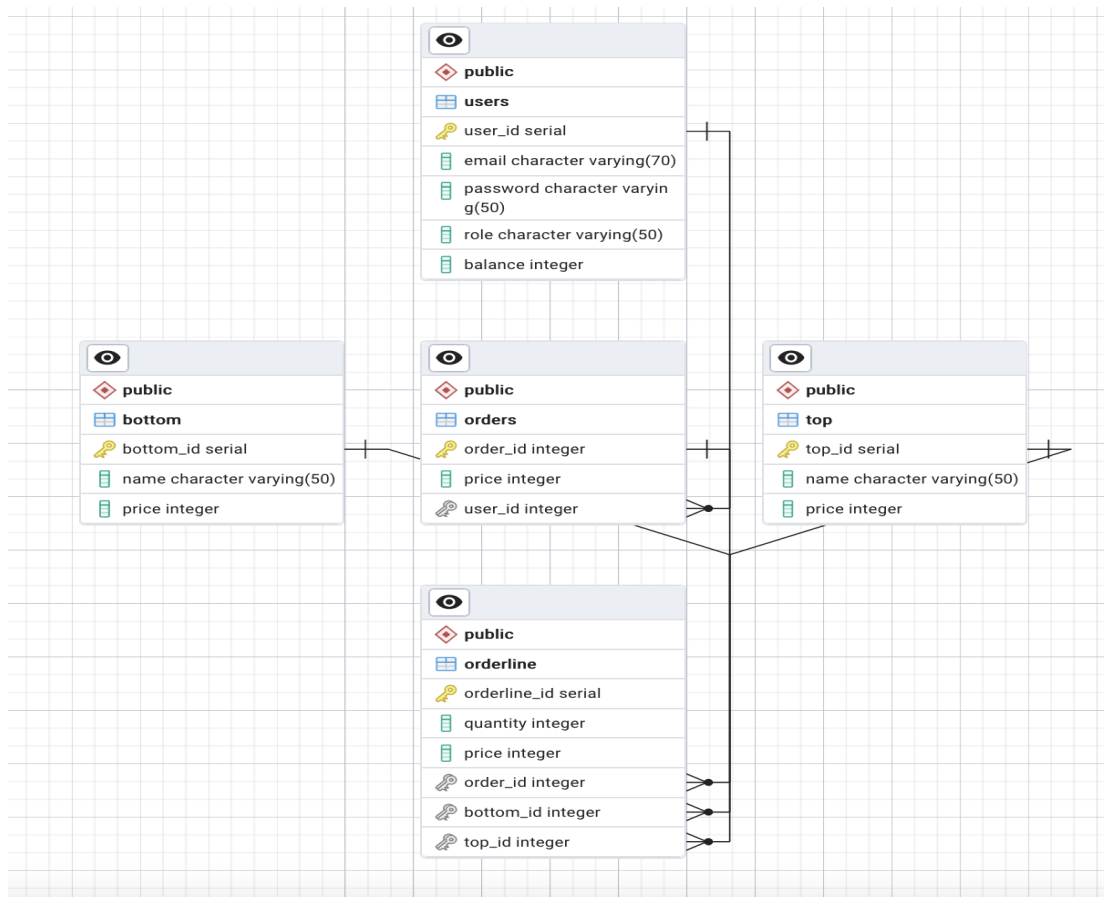
I vores ER-diagram er "top" og "bund" opdelt for at simplificere vores database og undgå at samle det hele i en tabel med navnet "cupcake". Dette valg blev truffet på baggrund af, at det er et skoleprojekt, og derfor er det ikke nødvendigt at inkludere yderligere oplysninger. Desuden er det kun cupcakes, vi har at gøre med, så vi regner ikke med, at de sælger wienerbrød, hvoraf det ville give mening at samle alt under en tabel, så man ikke behøver at have en tabel for hvert enkelt produkt, som butikken kunne sælge.

Ud over dette har vi sikret, at vores database opfylder de tre normalformer:

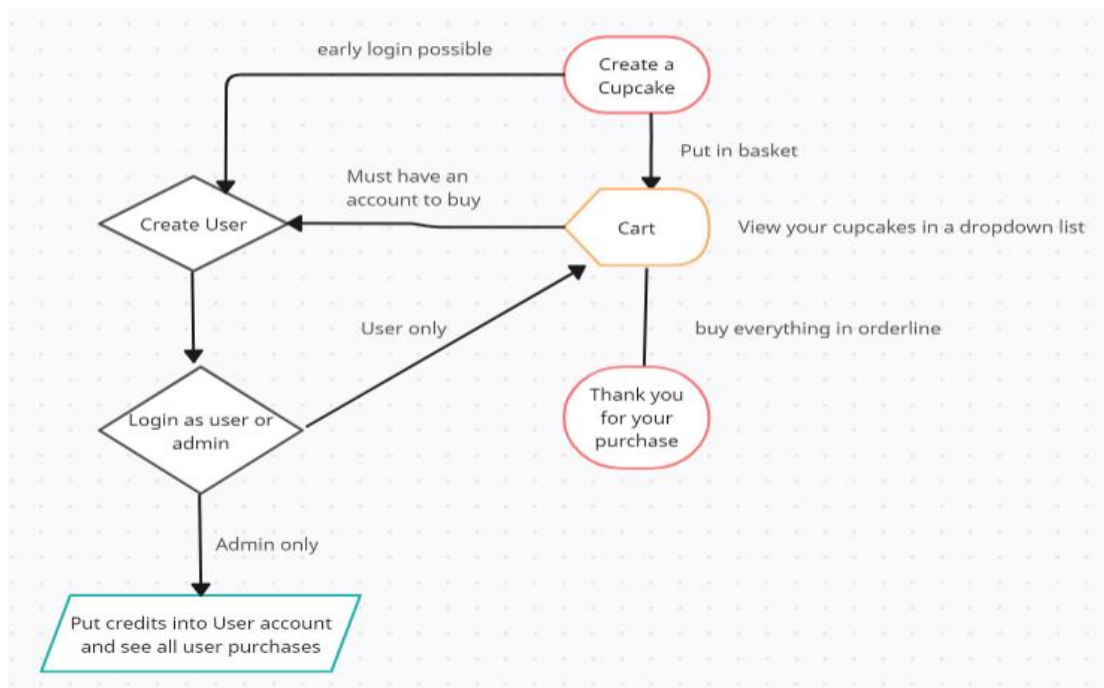
- 1: Ingen kolonner må gentage en anden kolonnes værdi.
- 2: Hvis en tabel har en sammensat nøgle, skal alle felter, der ikke indgår i nøglen, afhænge af den samlede nøgle.
- 3: Der må ikke findes felter uden for primærnøglen, som er indbyrdes afhængige¹.

Dette kan ses i diagrammet nedenfor.

¹ <https://balslev.io/programmering/database/normalisering-af-databaser/>



Navigationsdiagram



Særlige forhold

For at gøre hjemmesiden så enkel og effektiv som muligt, blev der taget et valg om, at kunden ikke skal kunne betale før denne er logget ind og virksomheden har navn og email på personen. Til det har vi lavet en smart if-else der viser en text som beder kunden logge ind hvis de endnu ikke er det, eller viser en køb knap hvis de er logget ind.

```
<div class="buy-login">
  <div class="buy-button">
    <div th:if="${session.currentUser != null}">
      <button class="payButton" type="submit" name="pay" formaction="/pay">Pay using wallet</button>
    </div>
  </div>
  <div class="need-login-text">
    <div th:unless="${session.currentUser != null}" class="logIntoPurchase">
      <p>You need to log in to make a purchase</p>
    </div>
  </div>
</div>
```

I overensstemmelse med User Story 5 ønsker Olsker Cupcakes at sikre, at brugeren altid kan se sin mail når de er logget ind. Vi tænkte lidt videre og kunne godt tænke os at anvende den samme login-side for både brugere og administratorer. Hvis en bruger har rollen som administrator, skal de omdirigeres til en administrations-side med et overblik over virksomheden efter vellykket login. Som man kan se på billedet nedenunder var vores tankegang at vi i UserController klassen ville lave en if statement i login metoden som tjekker om brugerens rolle er "admin".

```
if (user.getRole().equalsIgnoreCase( anotherString: "admin")) {
    ctx.redirect( location: "admin.html");
} else {
    ctx.render( filePath: "cart");
}
```

Når man så er logget ind ønsker Olsker at man altid kan se sin mail. Vi fandt ud af at vi kunne lave en if-unless statement i html ved hjælp af Thymeleaf. vores tilgang gemmer vi brugeren som en sessionattribut for at kunne tilgå deres e-mail. Derefter viser vi brugerens e-mail i et <p>-tag på hjemmesiden. Vi anvender et unless-statement for at vise en knap til at logge ind, hvis brugeren ikke allerede er logget ind.

Når admin.html-siden renderes ved tryk på knappen "add credit", indsættes der penge til brugerens balance. Der gemmes data fra ordrerlinjerne i en sessionsattribut, så admin.html-siden renderes med data fra orderlines.

```
<tbody>
<tr th:each="orderline: ${session.orderlines}">
... <td th:text="${orderline.orderlineId}"></td>
... <td th:text="${orderline.quantity}"></td>
... <td th:text="${orderline.price}"></td>
... <td th:text="${orderline.email}"></td>
</tr>
```

En sidste bemærkning er vores tilrettelæggelse af orderline, cartline og order. Siden vi i starten af projektet ikke har kunne gennemskue præcis hvordan disse skulle anvendes eller hvor data skulle gemmes.

Status på implementering

Som gruppe har vi opnået det meste af det, vi havde sat som mål. Der er dog nogle ting, som vi ikke fik gennemført. En af dem er at implementere en slet-knap, så køberen kan fjerne en ordre. En anden ting er, at vi har glemt at style vores HTML-side til kvitteringer. Der er heller ikke blevet implementeret CRUD-operationen DELETE i nogle af vores mapper, hvilket betyder, at vi mangler denne funktionalitet for at kunne slette en ordre.

Logout-knappen blev også overset (det ville bare at lave en en button og ctx.render den til forsiden men vi havde ikke mere tid), muligvis fordi vi havde større fokus på vores userstories, hvilket betød, at vi glemte lidt af hjemmesidens funktionalitet. Vi opdagede en fejl på vores hjemmeside: Hvis en bruger vælger en "top" og "bund" i starten og derefter går til adminsiden eller trykker på tilbage-knappen og forsøger at vælge en ny top og bund, vil vores site crashe. Dette kan være relateret til vores sessionattributte, men vi er ikke helt sikre på, hvorfor det sker, da vi ikke havde nok tid til at teste det.

Ellers har vi testet de fleste af vores metoder for fejl, som f.eks, at købere ikke kan købe en cupcake, hvis de ikke har nok penge på deres konti.

Derudover har vi ikke nået at få vores administrator-version af siden færdiggjort. Dette inkluderer muligheden for at man som admin kan trykke på sin e-mailadresse, for at nå til admin.html.

Proces

Et vigtigt aspekt af processen er, at vi som team har mødtes på skolen hver eneste dag i ugedagene op til deadline.

Den første dag besluttede vi os for et specifikt ambitionsniveau, der passede alle medlemmer af teamet. Med tidligere erfaringer i tankerne enedes vi hurtigt om at stræbe efter en høj standard og skabe et resultat, som vi alle kunne være stolte af. Vi havde tidligere oplevet problemer med GitHub i vores arbejde, så vi havde besluttet at sætte os ordentligt ind i det denne gang. Desværre endte vi med at skabe problemer for os selv. Vi indså, at vores repository var blevet oprettet forkert, hvilket potentielt kunne skabe problemer senere hen. En af grundene til at vi kunne arbejde igennem vores problemer var vores kommunikation og planlægning. Vi havde fra start besluttet at et Kanban Board ville gøre os godt, og vi gik flere gange tilbage til boardet, som et redskab der kunne hjælpe os med arbejdsgangen og flowet. Undervejs i projektet reflekterede vi løbende over vores arbejdsproces og identificerede områder, hvor vi kunne forbedre os.

Samlet set har vores arbejdsproces været bygget på kommunikation, planlægning og en samarbejdende tilgang til projektet. Selvom vi stødte på udfordringer undervejs, lykkedes det os at håndtere dem effektivt gennem en kombination af fejlhåndtering, kommunikation og læring.

Link til Youtube

<https://youtu.be/Amwr32dpEHU>

<https://youtu.be/ga56T4OioO8>