Cupcake projekt

Dette projekt er udarbejdet af gruppe 9 fra hold A Lyngby bestående af følgende. gruppemedlemmer:

- Kamilla Stoltenberg, cph-ks433@cphbusiness.dk, KamiStolt
- Alfredo Moises Fernandez N. , cph-af201@cphbusiness.dk AlfredoFernandez98
- Malte Pavón Olesen, cph-moo300@ cphbusiness.dk, Zeudiac
- Alberte Vallentin, AlberteVallentin, cph-av169@cphbusiness.dk

Indhold

Indledning

Dette projekt omhandler udarbejdelsen af en online platform til bageriet, Olsker Cupcakes. Vores opgave som udviklere er at realisere deres vision om et online platform, hvor kunderne kan bestille deres cupcakes, og hvor administratorer kan håndtere kundernes konti. Projektets formål er at vise vores evne til at arbejde med en virksomhed samt vores færdigheder inden for kodning, gruppearbejde og arbejdsproces. Vores mål er at levere en funktionel webshop, der lever op til virksomhedens krav. Rapporten er skrevet til og skal kunne forstås af en studerende på andet semester af datamatikeruddannelsen.

Baggrund

Olsker Cupcakes er et bageri, der er specialiseret i at lave dybdeøkologiske cupcakes. Virksomheden ønsker at udvide deres forretning til en online platform, hvor kunder kan bestille ordrer direkte fra deres hjem.
Kundens krav til systemet er, at det skal være en brugervenlig og funktionel webshop. Dette inkluderer at deres kunder kan bestille en valgfri bund og en valgfri top. Kunden skal kunne oprette en profil, for nemmere betaling samt at gemme en ordre. Log ind skal ske med email og kodeord, hvor der ønskes at e-mailen bliver vist på siden efter log ind. Olsker Cupcakes ønsker en administrator side, hvor administratorer har mulighed for at se alle kunder samt sætte penge ind på deres konto. Virksomheden ønsker derudover, at man som kunde skal kunne se sine ordrelinjer i indkøbskurven for at få et bedre overblik over den samlede ordre samt totale pris. Her skal det også være muligt for kunderne at slette en ordrelinje.

Teknologi valg

I projektet har vi brugt følgende teknologier:

- Figma (skal vi skrive det)
- IntelliJ IDEA 2023.3.6
- PostgreSQL 42.7.2
- Java 17
- Javalin 6.1.3
- Maven 4.0.0
- Thymeleaf 3.1.2
- JDBC
- JUnit 5.10.2
- Docker

Krav - Malte

Vores vision er at skabe en oplevelse for vores kunder inden for cupcake-bestilling ved at kombinere brugervenlighed, fleksibilitet og kvalitet. Vi vil tilbyde en platform, hvor kunder nemt kan bestille og betale for deres cupcakes online med valgfri bund og top.

Ved at oprette en konto på vores hjemmeside skal kunderne nemt kunne gemme deres foretrukne ordre og hurtigt gentage tidligere køb. Vi ønsker at skabe et miljø, hvor både kunder og administratorer kan have tillid til pålideligheden af vores system.

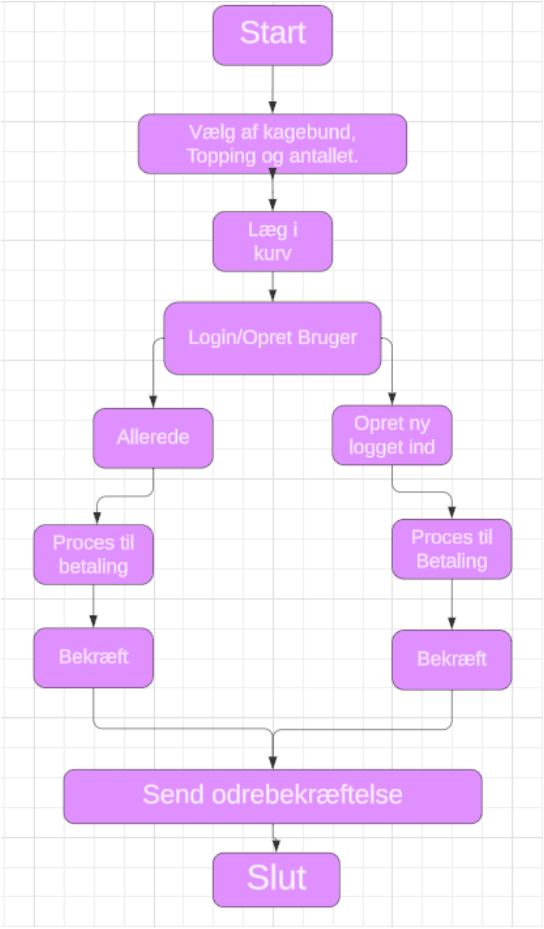
For administratorerne vil vi udvikle en administrations-side, hvor de kan følge op på alle ordrer og kunder i systemet. Dette giver dem mulighed for at holde styr på ordrer, administrere kundeinformation og sikre en problemfri drift af vores forretning.

Vores vision er at skabe en platform, der ikke kun gør det let og bekvemt for kunderne at bestille deres foretrukne cupcakes, men også giver administratorerne de nødvendige værktøjer til effektivt at styre og udvikle vores forretning.

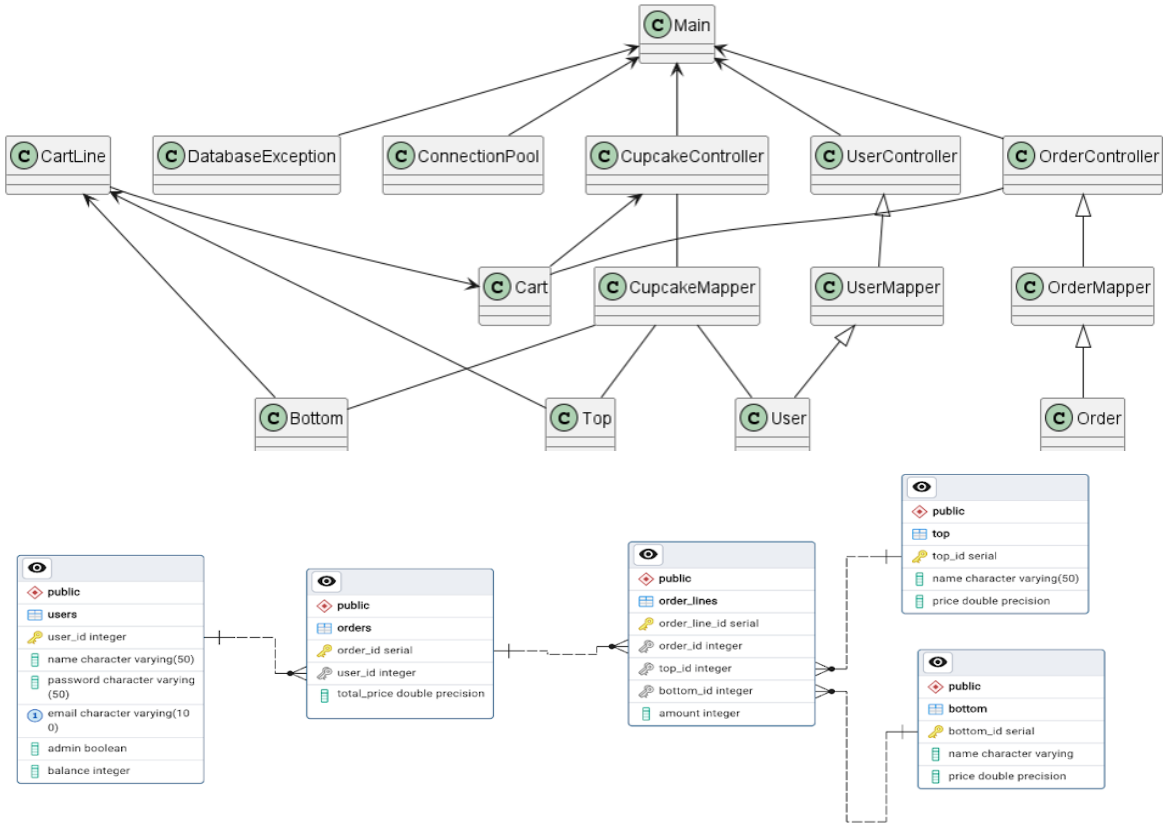
US-1: Som kunde kan jeg bestille og betale cupcakes med valgfri bund og top, sådan at jeg senere kan køre forbi butikken og i Olsker og hente min ordre.
US-2: Som kunde kan jeg oprette en konto/profil for at kunne betale og gemme en ordre.
US-3: Som administrator kan jeg indsætte beløb på en kundes konto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

1. **Vælg kagebund, topping og antal:** Kunden starter med at vælge kagebund, topping og antal som de ønsker at bestille. Dette trin giver kunden mulighed for at tilpasse deres bestilling.
2. **Læg i kurv:** Når valget er foretaget, lægger kunden de valgte cupcakes i deres virtuelle indkøbskurv. Dette trin tillader kunden også at fortsætte med at bestille, hvis dette er nødvendigt.
3. **Login/Opret bruger:** Kunden har mulighed for at logge ind på deres eksisterende konto eller oprette en ny konto. Dette trin er nødvendigt for at kunne fuldføre bestillingen af den valgte ordre, som er placeret i indkøbskurven.
4. **Opret ny logget ind:** Her får kunden mulighed for at oprette sig ind i systemet. Hvorefter bliver man sendt til indkøbskurv.
5. **Allerede:** Har kunden allerede en eksisterende konto, vil kunden blive sendt til indkøbskurv.
6. **Proces til betaling:** Efter man har trykket på "køb nu", så går kunden videre til bekræft.
7. **Bekræft/ ordrebekræftelse:** ved dette trin vil kunden modtage en bekræftelse, der angiver at ordren er behandlet og sendt afsted.



Domænemodel og ER diagram



Ovenstående billeder viser vores entity relation diagram samt domænemodel.

For vores projekt ville det have været godt at implementere en CartlineController i stedet for at have alle metoder i en CupcakeController og UserController.
Vi overvejede også, om vi overhovedet skulle have en top og bottom, hvor man i stedet kunne have et cupcakeobjekt.

Her kan det ses, at alle vores tabeller er på 3. normalform. Der findes kun en primærnøgle i hver tabel. Alle primærnøgler er autogeneret, og ingen felter uden for primærnøglen er afhængig af hinanden.

I vores users tabel har vi valgt, at email skal være unik, så to brugere ikke kan have samme email og dermed kan vi sørge for, at man ikke opretter en bruger med en email, der allerede er i brug, idet man i vores program skriver sin email og password for at logge ind. Derudover har vi en boolean, admin, der bruges til skelne mellem almindelige brugere og administratorer.

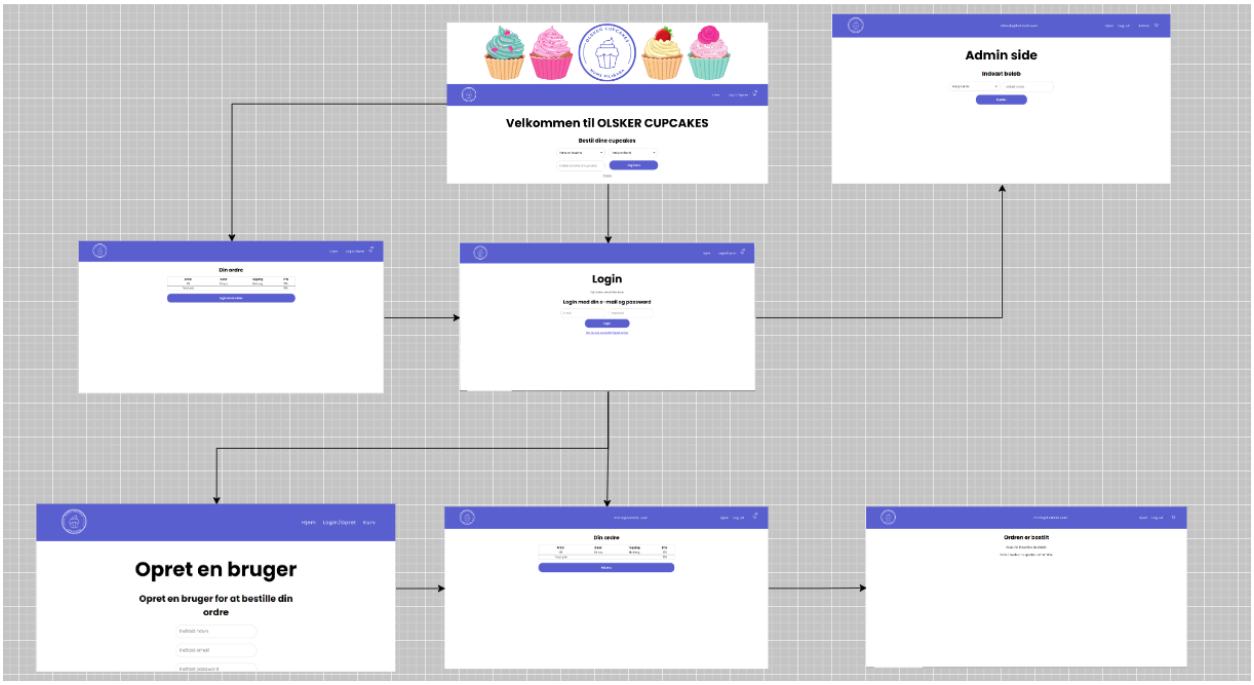
Vores orders tabel har en user_id som fremmednøgle og en mange-til-en-relation mellem disse to tabeller, så dermed kan én specifik bruger være relateret til mange forskellige ordrer, mens én specifik ordre kun kan være relateret til én bruger. Vi har derudover valgt, at price (både i top, bottom og orders tabellen) skal være en double, så der er mulighed for, at man kan ændre prisen til et kommatal.

Vores order lines tabel implementerer en mange-til-mange relation mellem orders og top og bottom tabellerne via fremmednøglerne order_id, top_id, og bottom_id.
Vores top og bottom tabel indeholder en liste over de henholdsvis mulige toppings- og bunde til cupcakes samt prisen på dem. Da vi kun har med salg af cupcakes at gøre, har vi valgt at lave de to tabeller. Skulle udvides til at udvides til at håndtere flere bagværk, ville vi fx implementere en tabel med bakery items, der ville fungere som en overordnet kategori for alle bagværker med attributten type, der ville angive, hvilket bagværk der er tale om.

Vi har valgt at have not null constraints i alle vores felter pånær balance. Dette er blandt andet med til at sikre, at dataene er komplette og meningsfulde. Fx i users, top, og bottom tabellerne, hvor hver række repræsenterer henholdsvis en bruger, en cupcake topping, og en cupcake bund, sikrer not null constraints, at hver bruger har en navn, email, osv., og hver topping eller bund har en navn og et pris. Dette gør blandt andet, at datakvaliteten er høj samt brugeroplevelsen er bedre, idet man fx altid kan se både prisen og navnet på en topping og en bund.

Vores balance kræver ikke nogen værdi. Dette gør, at man fx kan skelne mellem nye brugere/ikke-aktive brugere (der ikke har foretaget nogle køb/indskud) og brugere, der tidligere har købt cupcakes. Fra et forretningsperspektiv kan det fx bruges til at sende specifikke kampagne e-mails til disse brugere. Hvis man derudover vælger, at adminbrugere kun kan integrere med systemet og ikke købe cupcakes, angiver det tydeligt, at en saldo ikke er relevant for disse brugertyper.

Navigationsdiagram



Al aktivitet på vores cupcake projekt starter på vores forside. På forsiden kan man lægge cupcakes i kurven samt navigere til kurv eller login-siden, som er lokaliseret i vores navigationsbar.
Navigationsbaren er gennemgående på alle cupcake hjemmesidens sider og kan altid hjælpe en til at komme tilbage til startsiden, til kurven samt log ind/opret bruger og log ud.
Når man står i kurven, skal man være logget ind, før man kan betale for sin ordre. På vores loginside kan man enten logge på med administrator eller customer login credentials.
Herfra er det også muligt at oprette en bruger, hvis ikke man allerede har en. Når man har oprettet sig, vil man blive sendt tilbage til loginsiden.
Hvis man logger på med administrator credentials vil man blive sendt direkte videre til vores administrator side, hvorfra man kan indsætte penge på alle brugernes kontoer.
Logger man på som costumer, vil man enten blive sendt til forsiden (hvis man ikke har lagt nogle cupcakes i sin kurv) eller tilbage til ens kurv (hvis man har lagt cupcakes i den), hvorfra man har mulighed for at gennemføre ens køb eller vende retur til start for at bestille yderligere cupcakes.

Særlige forhold

Vi implementeret en særlig funktion, hvor administratorer kan logge ind på en administrations-side. Her har de mulighed for at administrere brugernes konti ved at tilføje penge til deres balance direkte på hjemmesiden. Dette giver administratorerne kontrol over brugernes kontosaldoer og muliggør en mere fleksibel og effektiv styring af betalinger og ordrer på hjemmesiden. Denne tilføjelse sikrer også en mere strømlinet og brugervenlig oplevelse for både administratorer og brugere.

Status på implementation

Ud fra de 9 user stories har vi fået implementeret de 8 første. Som kunde kan man på forsiden af hjemmesiden trykke på to forskellige dropdowns og vælge henholdsvis en topping og en bund ud fra de givne toppings og bunde fra databasen. Derefter kan man skrive/vælge antallet af cupcakes og tilføje det til ens kurv (bliver gemt som en ordrelinje).

Man kan vælge at gå videre til kurven eller bestille flere cupcakes. Er man logget ind i forvejen kan man fra kurven vælge at betale eller bestille flere cupcakes. Her kan man også vælge at slette en ordrelinje samt se det samlet beløb. Er man derimod ikke logget ind, vil man først blive bedt om at logge ind/oprette en bruger, før man kan kan trykke køb. Vores navigationsbar i toppen af hjemmesiden kan også bruges til at føre til de forskellige undersider. Her kan man også se sin e-mail, når man er logget ind.

Hvis man ikke er logget ind, vil man som udgangspunkt blive ført til loginsiden, hvis man trykker login/opret. Herfra kan man så komme til opret-siden, hvis ikke man har en bruger. Vælger man at logge ind før man har lagt cupcakes i kurven, vil man blive ført til forsiden efter login ellers vil man blive ført til kurven.

Er man administrator vil man derimod blive ført til adminstrator-siden, som også vil blive vist i navigationsbarren (kun for administratorer). Vores logo fungerer også som et link til forsiden. Er man logget ind, vil login/opret i navigationsbarren blive erstattet med log ud. Fra adminstrator-siden kan man se saldoen på de forskellige bruger samt indsætte penge til deres “konto”.

Når man har bestilt sine cupcakes vil man få en besked om, at der er bestilt (og kan hentes om en time). Her vil man også kunne se ens navn. Vi har sørget for, at der bliver vist en fejlmeddelse, hvis man ikke har valgt både en top, bund og et antal. Derudover bliver der også vist en fejlmeddelse, hvis ens passwords ikke er ens ved oprettelse en bruger eller hvis e-mailen allerede findes (er i vores database). Man får også en besked, hvis man ikke har nok penge, når man prøver på at købe en ordre. I denne besked bliver ens saldo også vist.

Igennem hele vores website har vi en meget ensartet look. Vi har dog ikke fået sørget for, at websitet både fungerer for almindelige skærme og mobiltelefoner. Indtil videre fungerer den kun til en laptop.
Vi har implementeret de CRUD-metoder, der har været nødvendige i forhold til det omfang vi har lavet. Fx kan man ikke opdatere prisen på bund/topping, men det var heller ikke et krav. Vi har heller ikke haft tid til at lave unit-test (som dog heller ikke var et krav).

Dog fik vi lavet en brugertest på tre personer. Her sad hver deltager alene med en observatør, hvor de blev bedt om at bestille minimum fire cupcakes med topping. Generelt var det let for brugerne at navigere på siden. Dog blev de alle forvirrede, da de skulle logge ind, fordi de ikke havde oprettet en bruger først. Da de så skulle bestille deres cupcakes, kunne ordren ikke gå igennem, da der ikke var sat penge ind på deres konto.

Proces

Vi startede vores forløb ud med at mødes fælles på skolen, hvor vi lavede en domænemodel og et klassediagram samt snakkede om, hvordan vi gerne ville have, at hjemmesiden skulle se ud visuelt, og hvordan databasen skulle struktureres. Hver aften har vi holdt et Discord møde, hvor dem der har haft mulighed for at deltage, var med. Her gik vi igennem, hvad vi havde lavet, debuggede og generelt aftalte, hvad den videre plan var. Vores database har haft små ændringer gennem forløbet, men "kernen" af den har forblevet det samme.

Fremover vil vi dog sørge for, at databasen (helst) er fastlagt fra begyndelsen, da det vil gøre alt andet meget lettere. Efter vores database og mockup i Figma var færdige, uddelte vi de forskellige user stories. Nogle af dagene har vi mødtes på skolen, mens andre dage har vi taget den hjemmefra. I starten virkede projekt lidt overvældende, og det var svært at finde hoved og hale i det hele, men efter vores database kom helt op og køre og forsiden var blevet kodet, begyndte det at skride fremad.

Det vi vil tage med til næste projekt er at sørge for, at fundamentet er helt på plads til at starte med. Det betyder, at vi har lavet et klassediagram samt domænemodel og database, som (helst) ikke skal ændres i. Derudover skal vi have et overblik over hvilke controllers og mappers vi skal have. I starten af dette projekt kodede vi stort set alt i en CupcakeMapper samt CupcakeController, hvilket hurtigt blev meget uoverskueligt. Da vi lavede nogle flere mappers og controllers, gav det et meget større overblik. Vi fandt også ud af, at nogle user stories var afhængige af andre.

I næste projekt vil vi derfor også kode fx vores userklasse (samt andre klasser, der bruges i en stor del af koden) til at starte med. Gennem forløbet har vi også holdt små Discord-møder og hjulpet hinanden på disse. Til næste gang vil vi sørge for, at vi for starten snakker om, hvordan folk arbejder bedst og dermed lave en aftale om, hvordan forløbet skal se ud, så det passer bedst muligt på alle. Derudover vil vi holde fast i daglige møder, hvor man kan catche up samt sørge for, at lave nogle faste deadlines, som skal overholdes.

YouTube-video (gennemgang af hjemmesiden)

Her er et link til en gennemgang af vores færdige produkt:
https://www.youtube.com/watch?v=l2lNpbv-fb0