

Cupcake - projekt



CPHbusiness Datamatiker Uddannelsen / 2. semester / Hold B / Gruppe 3

Afleveres: d. 11/4 - 2024

Udarbejdet af:

Sander Marcus Christensen / cph-sc296@cphbusiness.dk / Github: sanderMarcusChristensen

Marcus P Hjorth Rasmussen / cph-mr634@cphbusiness.dk / Github: MarcusHjorth

Mateen Jan Rafiq / cph-mr645@cphbusiness.dk / Github: MateenCPH

Indholdsfortegnelse

| | |
|--------------------------|----|
| Indledning | 2 |
| Baggrund | 3 |
| Teknologivalg | 3 |
| Krav | 4 |
| Aktivitetsdiagram | 5 |
| Domæne Diagram | 6 |
| ER diagram | 7 |
| Navigationsdiagram | 8 |
| Mock-up lavet i Figma | 9 |
| Særlige forhold | 10 |
| Status på implementation | 11 |
| Proces | 12 |
| Bilag | 13 |

Indledning

Dette projekt er et prøveprojekt på Datamatikerstudiet, hvor vi i forskellige grupper af 2-4 personer skal løse en opgave for et fiktivt bageri på Bornholm ved navn Olsker Cupcakes. Olsker Cupcakes ønsker at ændre det fysiske butiksflow ved at implementere pre-order til butikken. Bageriet ønsker en online løsning, hvor folk kan bestille og derefter afhente de valgte cupcakes. Olsker Cupcakes har opsat en række user-stories, som vi vil forsøge at imødekomme.

Rapporten skrives af datamatikerstuderende til datamatikerstuderende. Rapporten henvender sig til personer med lignende erfaring og viden. Læserne skal kunne forstå vores tanker bag projektet og forstå, hvorfor vi har truffet de beslutninger, vi har.

Baggrund

Olsker Cupcakes er et lille lokalt bageri i Olsker på Bornholm. Olsker er en lille by med 67 indbyggere (data fra 2009). Bageriet ønsker en online løsning, hvor de kan øge deres omsætning og ændre deres butiksflow.

Teknologivalg

Som valg af teknologier bruger vi IntelliJ IDE (v.2023.2.2) som hovedprogram til kodning. Databasen med tabeller er oprettet i PostgreSQL, og de grafiske mockups er blevet lavet i Figma. Selve projektets flow er dokumenteret i denne rapport. Til løsning af projektet er der også blevet anvendt et Trello-board samt GitHub's egen portal til løbende projektstyring. Nedenstående tabeller giver et visuelt billede af alle teknologier, værktøjer, biblioteker og buildtools, som er blevet brugt i dette projekt.

| Tools | Java libraries | Buildtools |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> • IntelliJ IDEA (2023.2.2) • PostgreSQL • pgAdmin • Github desktop • Figma desktop | <ul style="list-style-type: none"> • io.javalin.Javalin • io.javalin.rendering.template.JavalinThymeleaf • io.javalin.http.Context • org.thymeleaf.TemplateEngine • org.thymeleaf.templateresolver.ClassLoaderTemplateResolver • java.util.List • java.util.ArrayList • java.util.logging.Logger • java.util.logging.Level • java.sql.SQLException • java.sql.ResultSet • java.sql.PreparedStatement • java.sql.Connection • com.zaxxer.hikari.HikariConfig • com.zaxxer.hikari.HikariDataSource | <ul style="list-style-type: none"> • IntelliJ IDEA • Maven |

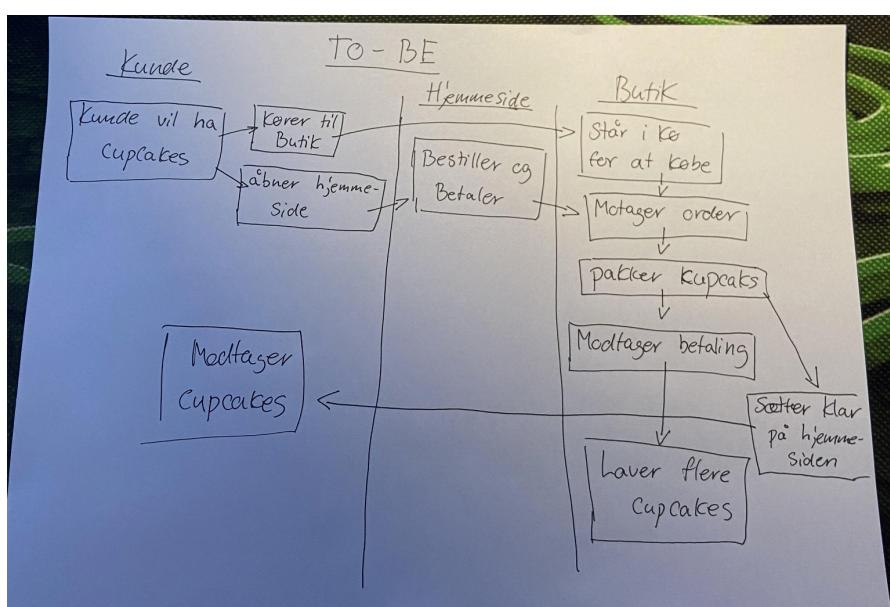
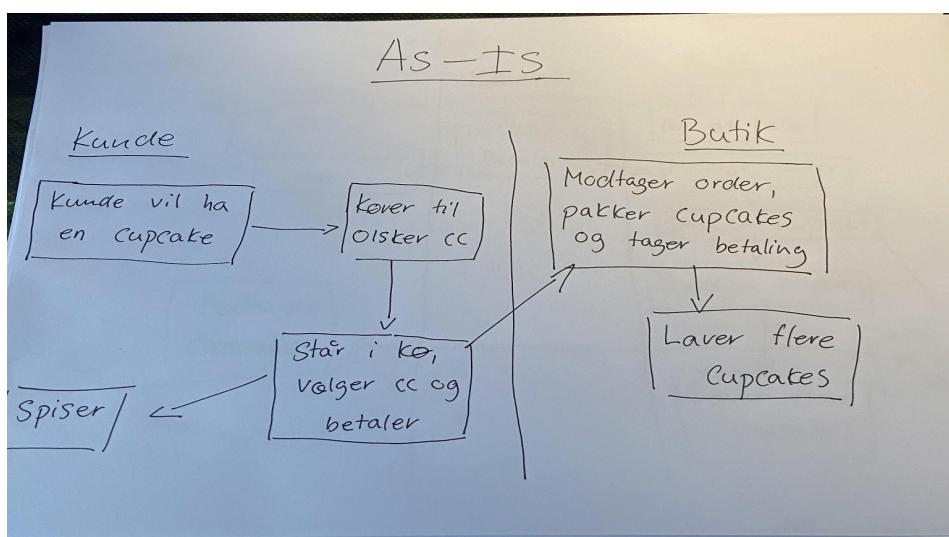
Krav

Butikken ønsker at indføre online pre-order til deres daglige drift. Olsker Cupcakes har selv opsat 9 forskellige user-stories som deres funktionelle 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 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

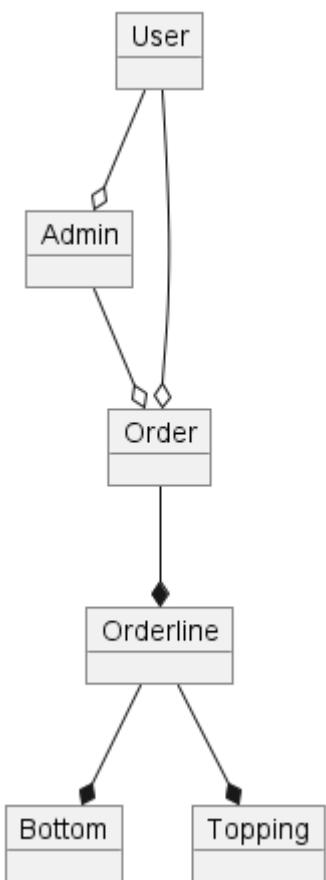
Det, som vores AS-IS/TO-BE-diagrammer beskriver, er, hvordan Olsker Cupcakes' nuværende butikssituation ser ud i forhold til, hvordan de ønsker, den skal være. Olsker ønsker at implementere flere trin, der automatiserer deres bestillingsproces, betalinger og generelle butiksflow. De håber, at denne hjemmeside vil kunne hjælpe dem med at sælge flere cupcakes dagligt og gøre salget af hver cupcake nemmere for deres kunder. Hjemmesiden skal give deres kunder mulighed for at bestille og betale hjemmefra samt have mulighed for fysisk ekspedition. Hvis 'Olsker Cupcakes' markedsanalyse holder stik, vil denne hjemmeside hjælpe med salg af cupcakes, på daglig basis.



Domæne Diagram

Sådan forestiller vi os, at domænet ser ud på den kommende hjemmeside for Olsker Cupcakes. Hjemmesiden kræver, at man har en bruger (User); uden en bruger ville man ikke kunne sætte navn på de cupcakes, der er blevet solgt. Vores (User) har muligheden for at være en "Admin", en bruger, som har flere rettigheder på hjemmesiden, samt adgang til flere informationer. Dette kan være nyttigt, hvis butikken løber tør for en bestemt cupcake og skal ændre dens status på hjemmesiden. De forskellige typer af brugere skal kunne lægge en stor

ordre (Order), bestående af flere forskellige typer cupcakes (OrderLine). Hver cupcake består af en top og en bund (Bottom and Topping).

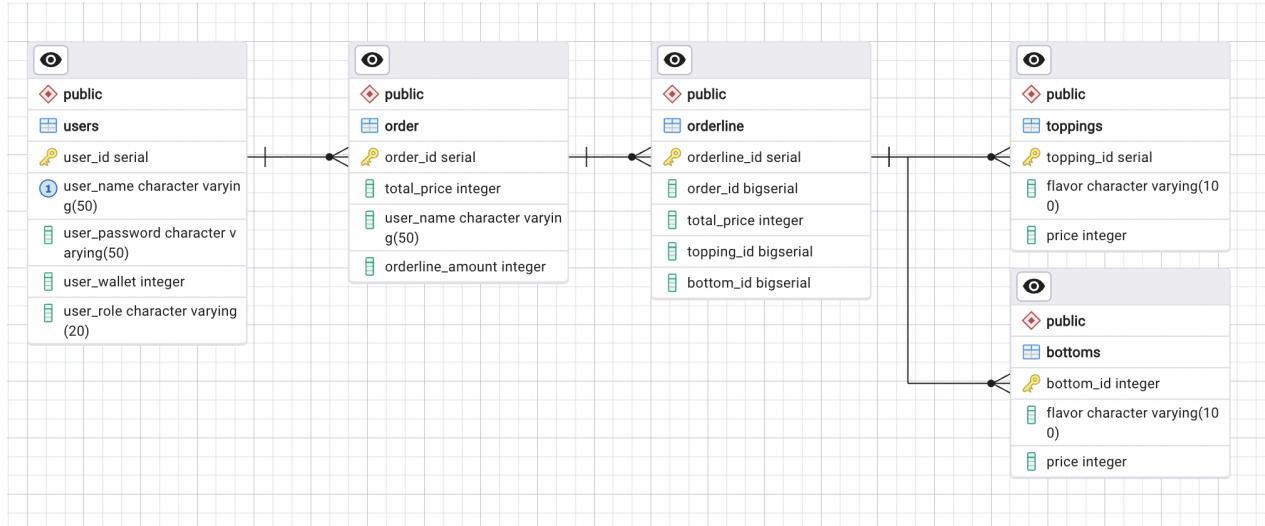


Relationerne mellem User, Admin og Order er ikke "stærke" relationer. Det skal være muligt at oprette/have en "User" uden at skulle være Admin eller lave en "Order" med den samme.

Relationerne mellem Order og Orderline er stærke, fordi en Order ikke kan eksistere uden en Orderline. Man kan ikke købe noget uden først at have bestilt det.

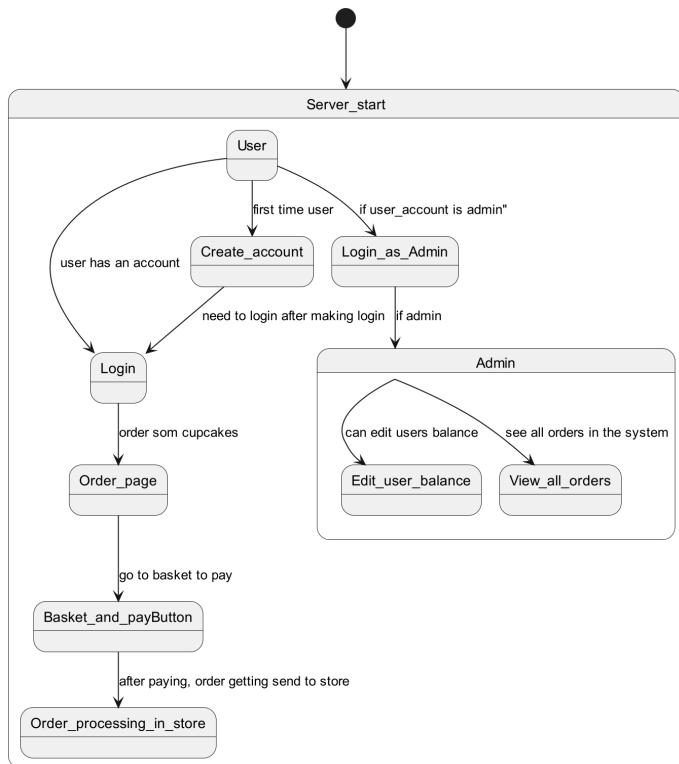
Orderline har stærke relationer til Bottom og Topping, da man kun kan bestille cupcakes hos Olsker Cupcakes. En cupcake består af en top og en bund, så det er derfor umuligt (på dette tidspunkt) at lave en Orderline uden både en bund og en top.

ER diagram



Alle vores tabeller har en “en-til-mange-relation”, hvilket vi har implementeret. Vi mener, at én bruger kan have mange ordrer, som igen kan bestå af mange ordrelinjer, der i sidste ende kan indeholde mange toppings og bunde. På nuværende tidspunkt er det dog ikke muligt at blande flere toppings og bunde hos Olsker Cupcakes. Vi mente dog, at det var bedst at fremtidssikre systemet i tilfælde af, at Olsker Cupcakes vælger at tillade flere toppings eller bottoms samtidigt i fremtiden.

Navigationsdiagram



Sådan forestiller vi os flowet på vores hjemmeside: Man skal enten logge ind eller oprette en konto, hvis man ikke allerede har en.

Efter man har logget ind, bliver man ført hen til bestilling af cupcakes. Man vælger hvilke cupcakes man ønsker og lægger dem i sin kurv.

Når man trykker på sin kurv, skal man kunne se alle de cupcakes, man har valgt. Her bør der være en knap til at gennemføre betalingen. Efter betalingen sendes orden til Olsker Cupcakes, så de kan forberede dem.

I tilfælde af, at man logger ind på en 'admin konto', har man flere muligheder. Her skal man kunne se alle ordre i systemet samt refundere penge til kunderne.

alle ordre i systemet samt refundere penge til kunderne.

Figma mockup:

På side 9. vises de visuelle mockups, som er lavet i programmet Figma. Dette er planen for hjemmesidens udseende samt alle dens funktioner.

Fremtidens tanke med hjemmesiden:

Vi tænker, at hjemmesiden skal kunne omdannes til en app, der fungerer på samme måde som "Starbucks" eller "Espresso House"-appen, i den nærmere fremtid. En app, hvor brugerne har en "wallet", hvori de kan indsætte penge og foretage betalinger gennem appen.

Mock-up lavet i Figma

Log In Page

Create account

HomePage / Orderpage

Basket

OrderPage

Display of users

Edit user balance

Sander M Christensen/ cph-sc296@cphbusiness.dk/Github: sanderMarcusChristensen
 Marcus P Hjorth Rasmussen / cph-mr634@cphbusiness.dk / Github: MarcusHjorth
 Mateen Jan Rafiq / cph-mr645@cphbusiness.dk / Github: MateenCPH

Særlige forhold

Hvilke informationer gemmes i session

Vi gemmer brugeroplysninger såsom id, navn, kode, rolle og deres wallet, samt hvad der lægges i kurven, selvom der ikke foretages en bestilling.

Hvordan håndterer man exceptions. Det kommer vi tilbage til senere i semesteret.

Det gør vi ved hjælp af database-undtagelser, try-catch-blokke samt fejlbeskeder til brugeren.

Hvordan man har valgt at lave validering af brugerinput.

For at oprette en ny bruger bliver der tjekket, om der allerede findes en bruger med samme navn som den, der forsøger at oprettes, i vores PostgreSQL-server, samt længden af både koden og navnet.

For at logge ind bliver brugerens input sammenlignet med de eksisterende brugere i PostgreSQL-serveren.

For at oprette en bruger, beder vi brugeren om at indtaste sit nye password to gange, således at brugeren ikke kommer til at lave fejlinput i deres password. Dernæst bliver disse to passwords sammenlignet, og hvis de stemmer overens, og brugernavnet også er godkendt, så bliver brugeren til sidst oprettet og smidt i databasen.

Hvordan man har valgt at lave sikkerhed i forbindelse med login

På grund af tidsmangel har vi ikke haft mulighed for at implementere sikkerhedsmekanismer. Hvis dette projekt havde været for en reel kunde, ville sikkerheden have haft højeste prioritet, og vi ville have anvendt hashfunktioner eller lignende til at kryptere adgangskoderne, i stedet for at gemme dem i tekstformat i databasen

Hvilke brugertyper, der er valgt i databasen, og hvordan de er brugt i jdbc.

Vi har to forskellige brugertyper: "users" og "admin". Brugere har adgang til siden som normale kunder, hvorimod administratorer har adgang til to ekstra sider: "orderPage", hvor man kan se alle ordrer for kunderne, og "viewUserPage", som giver administratorer mulighed for at se en liste over alle brugere.

Status på implementation

Vi har bestræbt os på at implementere alle brugercases, som Olsker Cupcakes har angivet. Dog er der nogle mindre fejl, og vi har ikke kunnet implementere dem alle på grund af tidsmangel. Vi er opmærksomme på, at vores program ikke er så "fleksibel", som det kunne have været. Hvis Olsker Cupcakes vælger at sælge andet end cupcakes i fremtiden, kan det kræve betydelige ændringer i programmet og dets opsætning.

- User-Stories 7-9 er ikke blevet implementeret på grund af tid og opgaveformulering.
- Som nævnt tidligere i "særlige forhold" er vores sikkerhed ikke i top. Vi "checker" kun på, om brugernavnet og password har en maksimal kapacitet på 50 karakterer. Der er ikke noget sikkerhed angående minimum, man kunne f.eks. lave en bruger med brugernavn: A og password :1
- Hver gang vi klikker på "lige i kurv", skal alt info om vores cupcakes loades fra databasen.
- Som admin kan jeg sætte penge ind på en "User wallet" i postgres, men kan ikke bruge mit "wallet" til at købe cupcakes med endnu.
- Mangler en "error page", en fejlsidse man kommer hen til, hvis noget går galt i programmet.

Proces

Vores arbejdsform og samarbejde har været rimelig gode siden "Lifehack 2024" (et andet større projekt på Datamatikeruddannelsen). Derfor valgte gruppen at tage imod denne udfordring som dette projekt udgør. Vores tilgang har været at mødes omkring kl. 10 hver dag og derefter afsætte et par gode timer til diskussion, skrivning og kodning.

Alle gruppemedlemmer har bidraget på tværs af forskellige opgaver for at øve sig i de basale komponenter i projektet. Alle har lavet mindst én HTML-side, skrevet kode til vores backend og bidraget til denne rapport. Dog er det meste af vores HTML og CSS blevet udarbejdet af Marcus, Mateen har været ansvarlig for vores Controller-klasser og vores Database. Sander har taget sig af den resterende backend, såsom Mapper- og Entitetsklasser, samt stået for udarbejdelse af diagrammer og dokumentation.

Gennem denne opgave har vi lært at kommunikere bedre som et team, selvom der stadig er plads til forbedring. Alt fra at spørge om hjælp til at diskutere og forklare kode til hinanden. Vi har også lært, at det er vigtigt at planlægge bedre, inden vi går i gang, og at et dagligt stand-up møde er en god idé. Desuden skal vi blive bedre til at begrænse os, da vi i denne opgave havde for mange opgaver i gang på én gang, hvilket resulterede i, at mange ting blev efterladt halvfærdige. Alle disse punkter bliver taget med videre til næste projekt, hvor vi forhåbentligt kan forbedre os på disse ting.

Vi har også erkendt, at brugen af Kanban-bordet er en god praksis, når det anvendes korrekt. Desværre glemte vi imod slutningen af opgaven at bruge vores Kanban-bord, hvilket var ærgerligt, da det giver et godt overblik.

Demo video

Bilag

- **Trello (Kanban-bordet):**

<https://trello.com/invite/b/yCWyq98Q/ATTlc882055397f207f8dac6b73087ce0b8aD0DD8449/newsbox>

- **Opgavebeskrivelse:**

<content/cupcake/README.md at main · dat2Cph/content · GitHub>

- **Git Repository med projekt :**

<https://github.com/sanderMarcusChristensen/Cupcake>