Cupcake

Deltagere

Klasse A, 11/04/2024.

Daniel Rouvillain - cph-dr136@cphbusiness.dk, Rouvii

Kevin Løvstad Schou - cph-ks427@cphbusiness.dk, cphkev

Mads Oliver Rosengren - cph-mr632@cphbusiness.dk, ScriptCodes

Matthias Sigurdsson - cph-ms1166@cphbusiness.dk, Mattsigurd.

Link til youtube:

https://www.youtube.com/watch?v=u6HJjDTRNeI&ab_channel=Rouvi

Indholdsfortegnelse

Indledning

Målgruppen til denne indledning er datamatiker-studerende på 2. semester.
Dette projekt går ud på at skabe en website for Olsker Cupcakes, hvori vi har som mål, at udføre 9 user stories. For eksempel at kunne skabe en konto eller bestille cupcakes med valgfri bund og top.
I denne rapport kommer vi ind på, hvad baggrunden for projektet er, hvilke krav opgaven stillede os, og så viser vi nogle modeller og diagrammer for at give en oversigt over indholdet og relationer i vores kode. I slutningen kommer vi ind på nogle særlige forhold i vores kode, hvilken status projektet har og hvordan processen i teamet har gået.

Baggrund

Baggrunden for dette projekt er firmaet Olsker Cupcakes. To hipsters fra København har været forbi bageriet, og har lavet en mock-up og nogle krav til vores team, og vil have at vi bygger en funktionel website, til at kunne bestille deres cupcakes.
De krav som blev sat for os var at kunder skal kunne bestille og betale for cupcakes, med en valgfri bund og top, så de senere kan hente deres ordre i butikken. Kunden skal også kunne skabe en konto, for at kunne gemme sine ordrer. Kunder skal kunne sætte ordrerlinjer i en indkøbsvogn, så de kan se den samlede pris. I denne indkøbsvogn skal man også kunne fjerne sine ordrelinjer, så de kan justere deres ordre, som de vil.
Man skal også som adminstrator være i stand til at styre ordrerne, ved at sætte et beløb ind i Postgres, så kunder kan betale, være i stand til at se ordrer i systemet og kunne se kunderne i systemet, så de kan følge op på ordrer os holde styr på kunderne.

Teknologivalg

Sprog: Java.

Byggesystem: Maven.

JDK: Correto-17.

Database: Postgresql.

Kode editor: IntelliJ 2023.3.5.

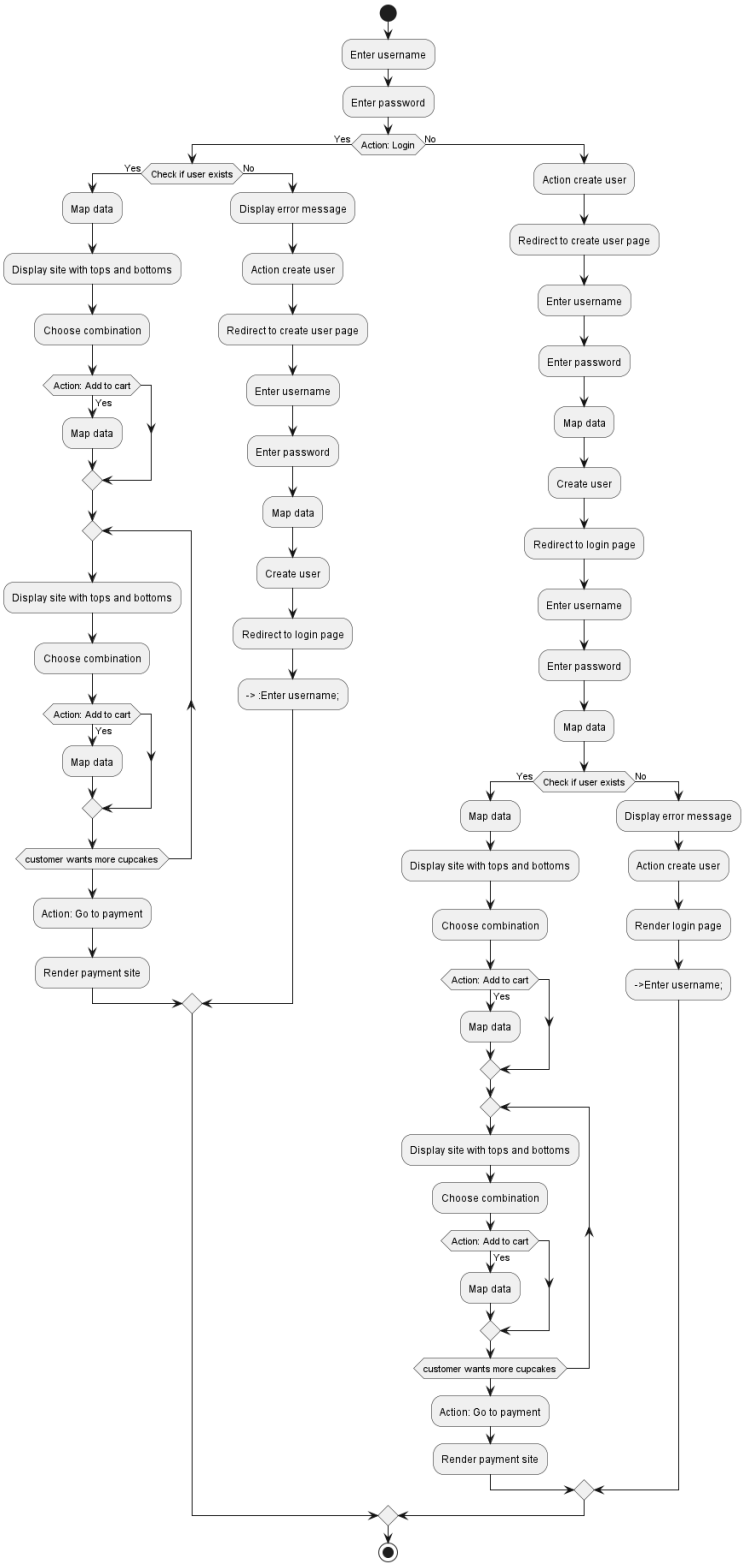
Krav

Firmaets håb er at få en funktionel website, som gør det nemmere for kunder at købe cupcakes og hente dem i butikken, men også gøre det nemmere for medarbejderne til at holde styr på kunderne og deres bestillinger.

Her ligger de krav der blev sat til websitet:

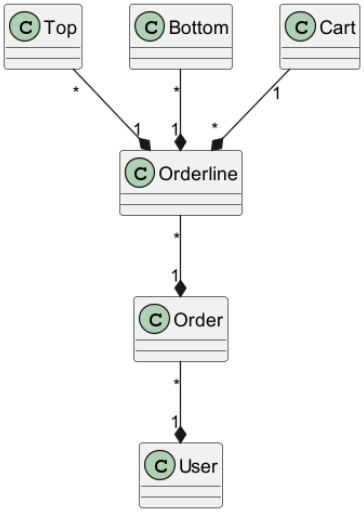
- **US-1:** Som kunde kan jeg bestille og betale cupcakes med en valgfri bund og top, sådan at jeg senere kan køre forbi butikken i Olsker og hente min ordre.
- **US-2:** Som kunde kan jeg oprette en konto/profil for at kunne betale og gemme en ordre.
- **US-3:** Som administrator kan jeg indsætte beløb på en kundes konto direkte i Postgres, så en kunde kan betale for sine ordrer.
- **US-4:** Som kunde kan jeg se mine valgte 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 mock-up'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.

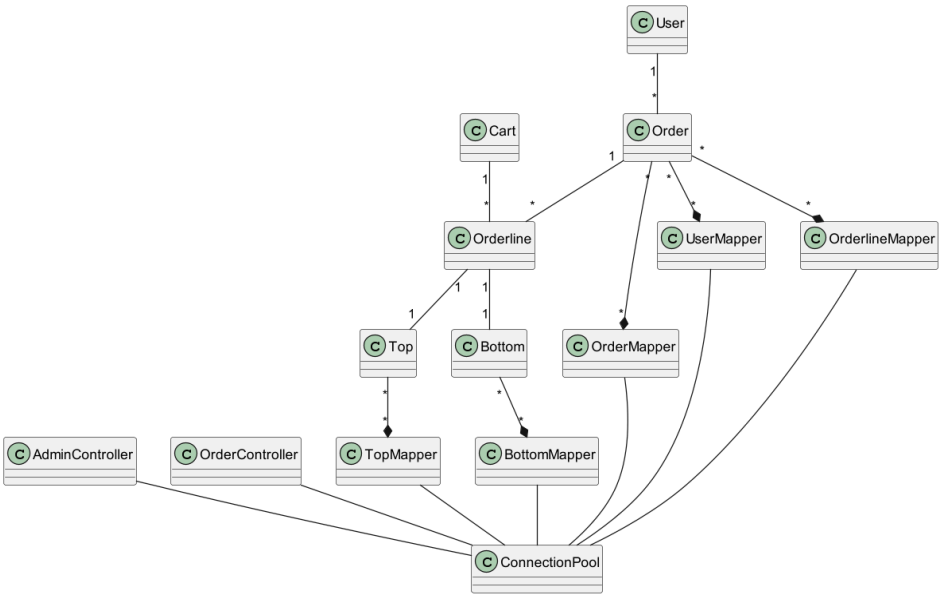
Aktivitetsdiagram



Domænemodel

Den første domæne model viser hvor vi startede, hvor den sidste domæne model viser hvad vi er endt ud med. Dette viser hvordan projektet har udviklet sig over de sidste par uger.





Først og fremmest er der en bruger(User) som i her er den primære aktør i systemet, det er dem der kan placere ordrer på forskellige cupcakes. De cupcakes som brugeren kan bestille består af to komponenter, altså en Top og en Bund.

Toppen består af forskellige typer fyld/pynt, som brugeren kan vælge at komme på sin cupcake, hvor bunden repræsenterer cupcake bunden.

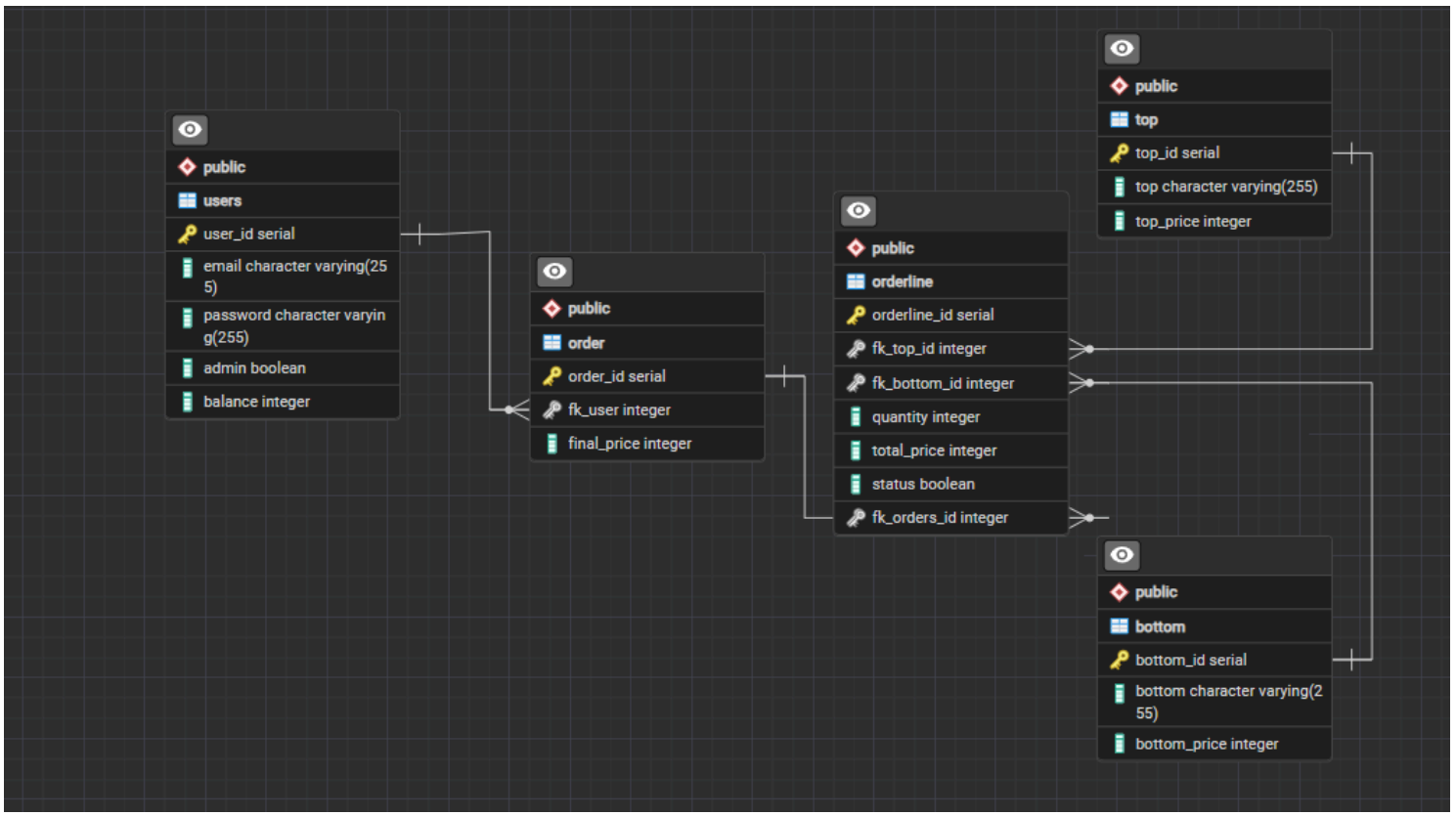
Når brugeren har angivet sine valgmuligheder, bliver der placeret en bestilling, hvilket opretter en ordre(Order) i systemet. Denne ordre "sammensætter" valgmulighederne fra brugeren og gør det til en cupcake.

Hver ordre bliver så repræsenteret af en ordrelinje(Orderline). Orderlinjen indeholder data om den specifikke top og bund som udgør denne cupcake, samt antallet af cupcakes af denne type cupcake.

Systemet har også en indkøbskurv(Cart) som er med til at gøre det nemmere for brugeren at

få overblik over hvilke cupcakes de har valgt, samt hvad den samlede pris er for den gældende cupcake.

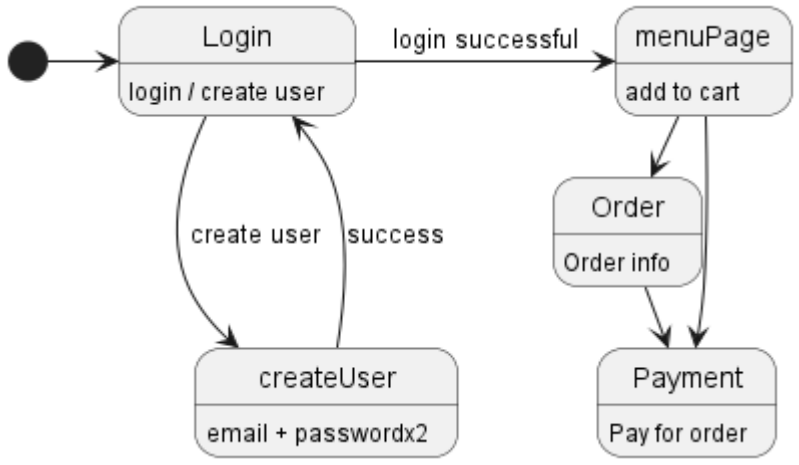
ER diagram



Databasens struktur og opbygning bygger på baggrund af flere overvejelser.
Pr. kundens anmodning, user stories, er det ikke en sædvanlig cupcake webshop, som sælger pre-lavet cupcakes. En af udfordringerne ved applikationen, og dermed databasen, er at en cupcake er sammensat af en valgfri top og bund. På baggrund af det valgte vi at lave to separate tabeller til toppe og bunde med hver deres PN, navn og pris.
Vi har senere henne i projektet spekuleret over om, det har været nemmere at lave en tabel, der samler bund og top, og på den måde lave kombinationerne, for at gøre det nemmere at

arbejde med.
Vi blev hurtigt klar over, at opgaven krævede en orderline, der består af henholdsvis bund og top som FK's. Da man skal kunne bestille mere end en af den sammensatte cupcake, er en mængde(quantity) derfor også nyttig. Ydermere vil vi gerne holde styr på, hvorvidt en orderline er betalt eller ej. Dog i skrivende stund, og efter nærmere eftertanke, er status entiteten mere tiltænkt til order tabellen, da vi i realiteten gerne vil tjekke om den enkelte ordre er betalt eller ikke.
Orderline tager hertil også et order som FK. Da vi gerne vil oprette en ordre til vores orderlines.
Order tabellen tager hertil en user som FK, da vi vil tage fat i en user og knytte den til den pågældende ordre, så de kan få fjernet deres balance.
User har de overordnede ting, som en PN, navn/email og password. Ydermere har den en entitet "admin", som er en boolean, da vi var klar over, at vi skulle operere med en admin side.

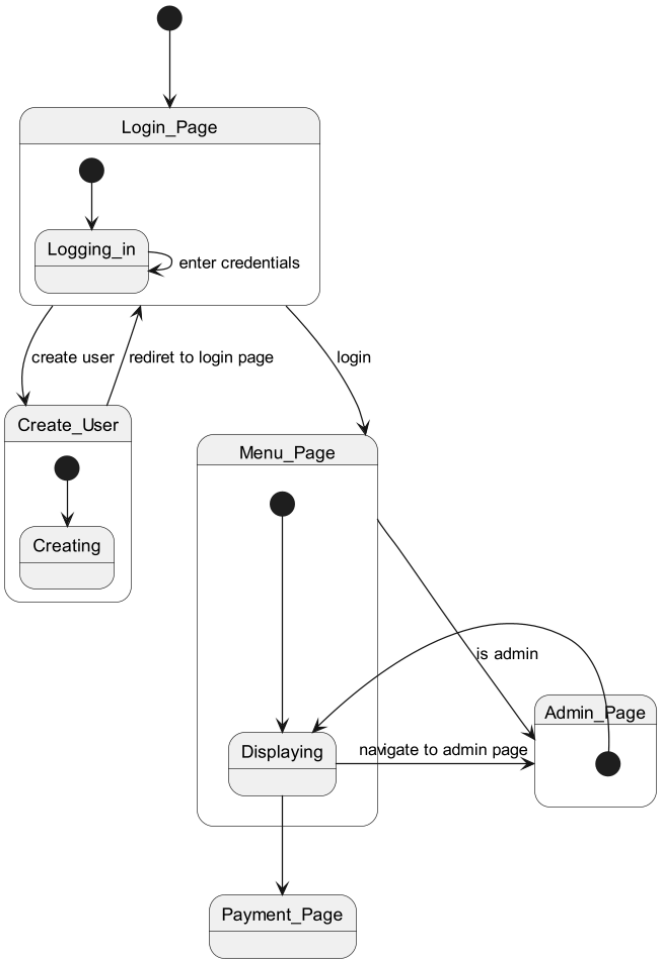
Navigationsdiagram



Kunden starter i login-siden, hvor man kan vælge at logge ind eller skabe en konto. Når man har logget ind succesfuldt, kommer man videre til menu-siden, hvor man kan vælge imellem cupcake bunde og toppe og føje dem til sin indkøbsvogn. Til sidst kan man enten kigge på sin ordre og gå direkte til betalingssiden.
Dette korte navigationsdiagram kunne være lidt større, hvis vi havde bygget videre på projektet. Vi ville også have en admin side, som var tilgængelig for brugere, som er logget

ind med admin status. Der ville man være i stand til at se kunderne og deres ordrer. Der kunne også have været en "payment complete" side, hvis ordren gik igennem systemet.

Oversigtsdiagrammet.



Særlige forhold

I et større projekt, kan man støde op mange exceptions, og derfor skal man kunne håndtere dem effektivt. Vi byggede en exceptions-package med en klasse der kan håndtere de forskellige exceptions vi støder på. Dette kan hjælpe med at give et bedre overblik og klarhed på hvilket slags exception man skal håndtere. Det kan også hjælpe med kodens genanvendelighed, og give en konsistent måde at håndtere problemer i koden.
For at have et login og bruger-system på en website, burde man have noget fokus på sikkerheden af disse konti. Så selvom man ikke kan lave en bruger, som allerede findes:

```
catch (DatabaseException e)
{
    ctx.attribute("message", "Dit brugernavn findes allerede. Prøv igen, eller log ind");
    ctx.render("createuser.html");
}
```

Kan man sætte krav til hvad dit kodeord kan være, så det er svært at hacke. Det kunne f. eks. være at have et minimum længde kodeord på 8 tegn. Krav om at der skulle være et specielt tegn, et stort bogstav, eller andet.

Status på implementation

- Mangler overordnet styling på alle html siderne.
- Mangler at lave US-8
- Mangler at tage data fra menupage.html med over til overview.html så kunden kan betale for sine vare
- Mangler at fikse fejl hvor den første vare der bliver tilføjet til indkøbskurv, ikke tilføjer til den totale sum af alle varerne i indkøbskurven.
- Mangler en metode der kan tage en admin til adminpage.html siden, samt at lave en knap på index.html siden for at gøre det nemt for admin.

- Mangler at fikse fejlen hvor du kan oprette uendelige mængder af brugere uden et username eller password.
- Mangler at indstille en max værdi på antal af cupcakes en bruger kan tilføje til kurv
- Hvor du tilføjer cupcakes til indkøbskurven og indkøbskurven bevæger sig samlet længere op ad siden nu flere gange du tilføjer elementer til indkøbskurven, dette resulterer i at man ikke længere kan trykke på "tilføj til kurv" samt "gå til betaling".

Proces

Projektet startede med en gennemgang af den forventede arbejdsindsats fra gruppen.
Der var derfor en klar forudsætning for gruppens arbejdsindsats, og hvad det krævede fra gruppen at nå til mål med et endegyldigt og enestående projekt.
Det var af den opfattelse, at alle gruppens medlemmer mødtes op fysisk i den første del af projektet. At møde fysisk op gav mulighed for at modtage den hjælp, der skulle være brug for.
Der var ikke en klar aftale om anden del af projektet.

Planen for projektforløbet er i sig selv ret klar; få lavet hele opgaven. Der var ikke sådanne planer over det pågældende projektforløb i vores gruppe, så det kan ikke beskrives.

Projektforløbet kan beskrives som rodet, da vi manglede struktur og organisering fra start. Kanban var et godt redskab for at skabe en form for struktur og overblik over gruppens arbejde, dette vil bruge meget mere til de kommende projekter.

Det der kunne have været bedre var gruppens kommunikation, da vi nok ikke fik uddelegeret ansvarsområder grundigt nok, hvilket har resulteret i at mange af metoderne ikke er kommet på plads i vores projekt. Dette vil vi tage til os, så vi til en anden gang kan arbejde mere struktureret i projektforløbet.

Dette projekt har alt i alt været en god læringsmulighed for os som en gruppe, da vi nu har en bedre forståelse af hvert gruppemedlems styrker og svagheder når det kommer til forskellige dele af kodebasen, samt hvor vigtig struktur kan ende med at være i så stort et projekt som dette.