Cupcake Projekt

24-03-2025 - 04-04

Team 6 - Hold B

Emil K.
cph-ek218@cphbusiness
github: emil2112

Thor
cph-ts401@cphbusiness.dk
github: ThorSchou

Markus
cph-mj1192@cphbusiness.dk
github: Bojuuh

Indholdsfortegnelse

Indledning

Dette projekt er udviklet som en del af datamatikeruddannelsen på 2. semester og henvender sig til medstuderende på samme niveau. Projektet går ud på at udvikle en fiktiv cupcake-webshop, hvor brugeren kan sammensætte sine egne cupcakes ved at vælge bund og topping, og derefter tilføje dem til en indkøbskurv og gennemføre en ordre. Formålet med projektet er at lære, hvordan man udvikler et fuldt funktionelt websystem med login, sessionsstyring, databaser og dynamisk HTML.

Baggrund

Dette projekt omhandler en fiktiv Cupcake forretning som skal have udviklet en webshop, hvor deres kunder kan forudbestille Cupcakes som kan afhentes i butikken på et senere tidspunkt.

Teknologivalg

Teknologi	**Version**
IntelliJ IDEA	2024.3.2.2
Java	17
Javalin	6.5.0
PostgreSQL	42.7.5
JDBC	4.3
Thymeleaf	3.1.3
Hikari	6.2.1
Docker	4.38.0

Krav

Vi har som udviklere fået udleveret et mockup af den hjemmeside som kunden ønsker, samt de nedenstående 9 user-stories som beskriver den funktionalitet som kunden ønsker der skal være på hjemmesiden.

Vores figma prototype baseret efter mockup fra kunden:

https://www.figma.com/design/MrIJXXKNGMF2FlDVF0moLu/Cupcake?node-id=0-1&t=6iJVT5RGXAI71Yab-1

User-stories:

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.

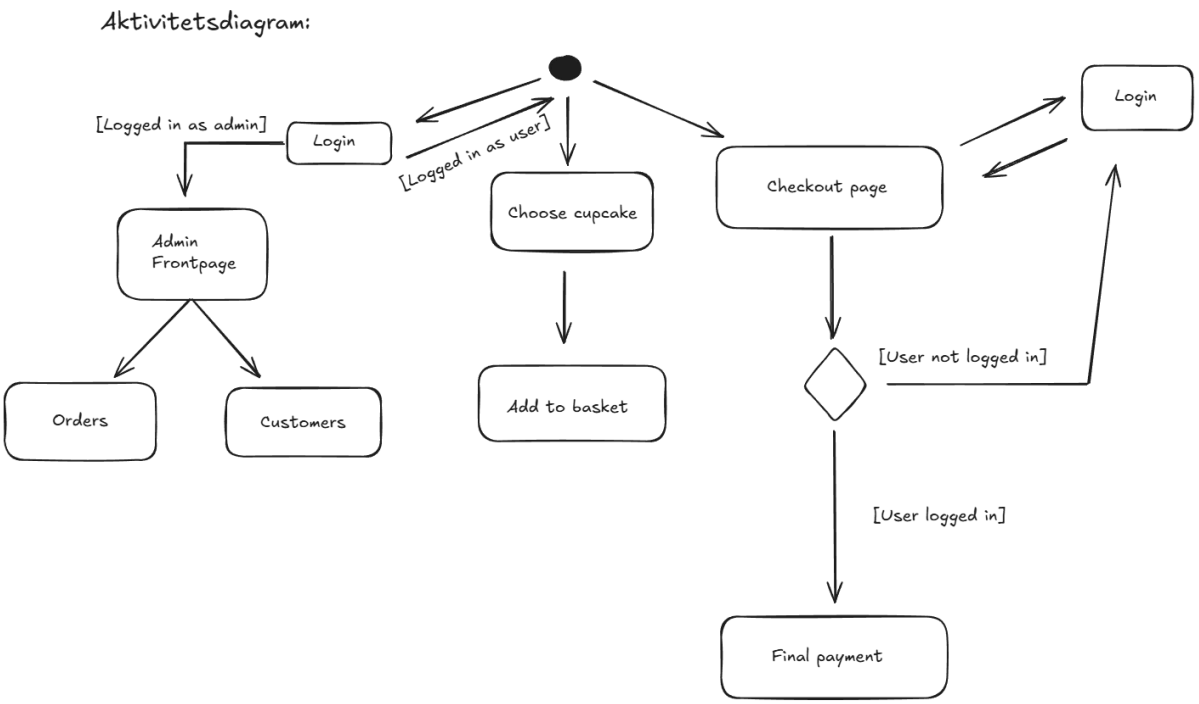
US-7: Som administrator kan jeg se alle kunder i systemet og deres ordrer, sådan at jeg kan følge op på ordrer og holde styr på mine kunder.

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.

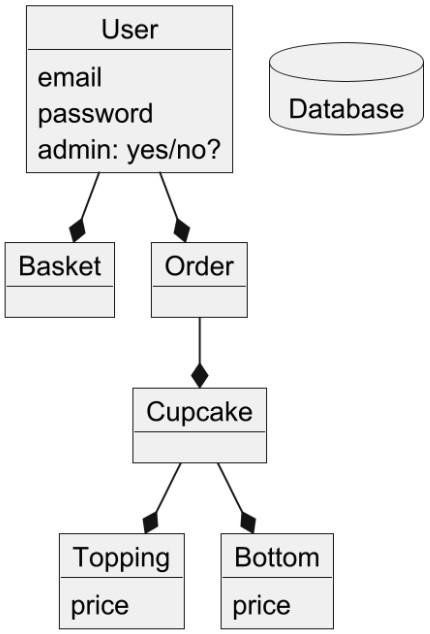
Aktivitetsdiagram

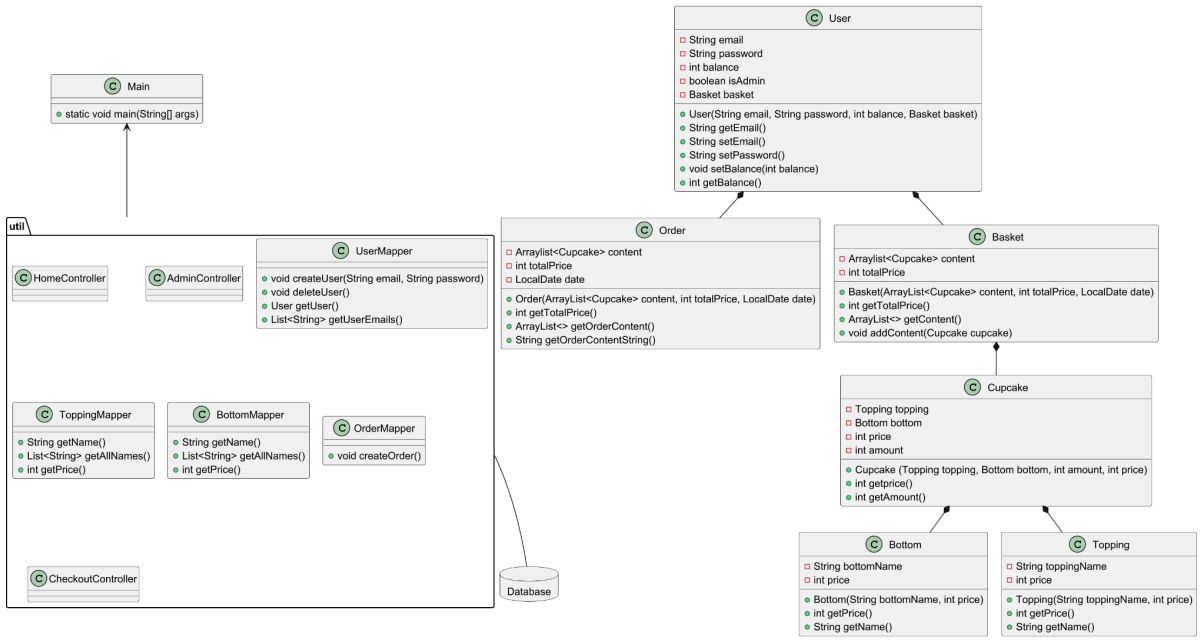
I begyndelsen af projektet udarbejdede vi et aktivitetsdiagram, for at skabe et overblik over det overordnede workflow i Olsker Cupcakes, og for give os en ide om hvordan vi ville bygge hjemmesiden.
Aktivitetsdiagrammet dannede grundlag for domænemodellen og vores klassediagram.



Vi brugte en side kaldet www.excalidraw.com til at skitsere aktivitetsdiagrammet. Det største emne ved udarbejdelse af diagrammet var hvordan vi skulle håndtere om brugeren var logget ind, og om de var logget ind som administrator.

Domænemodel og Klassediagram

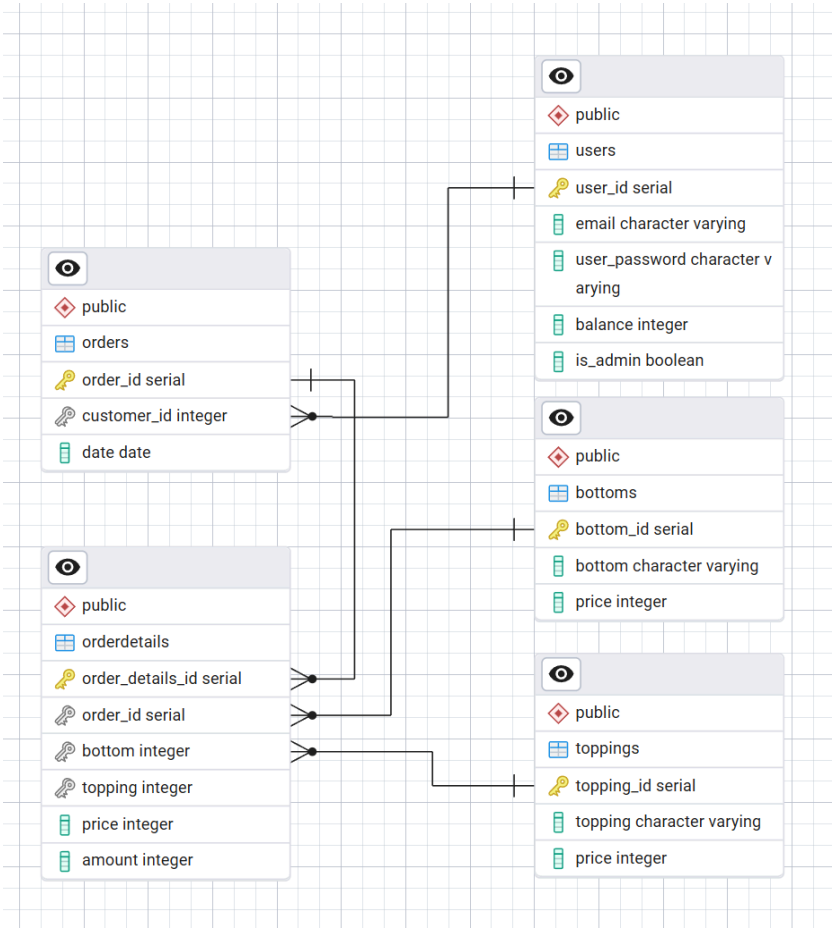




Vi overvejede, hvorvidt en bund og en top "*is a*" cupcake, og dermed skulle benytte sig af nedarvning fra Cupcake klassen. Vi besluttede, at de ikke var, da en bund og en top ikke kan være en cupcake i sig selv, men i kombination bliver de det. Altså at en cupcake "*has a*" bund og top. Af den grund benytter vi os ikke af nedarvning her.

En anden overvejelse vi havde var forskellen mellem Order og Basket. For at få det til at spille med databasen bedst, valgte vi at lave Basket til en midlertidig beholdning af cupcakes før køb, og en Order til at repræsentere et gennemført køb.

ER diagram



Vores ER diagram består af fem tabeller: users, orders, orderdetails, toppings og bottoms.

Users-tabellen indeholder kundernes oplysninger, orders registrerer deres køb og orderdetails fungerer som en forbindelsestabel, der kobler ordrer med forskellige kombinationer af bunde og toppings. Til sidst har vi toppings og bottoms-tabellerne som gemmer de forskellige smagsmuligheder og deres priser.

Normalisering og relationer

Vi har lavet vores ERD ud fra de tre normalformer. Vi afviger dog i vores orderdetails tabel, hvor vi har en "price" kolonne som er afhængig af andre attributter, men bruges til at vise totalpris på en ordre, til fremtidig brug.

Tanken er at hvis price hentes fra de tabeller som den er afhængig af, så ville ordre historikken kunne "ødelægges" af prisændringer i fremtiden.

Vores ERD bruger flere 1:M relationer. En User kan have flere Orders, men hver ordre tilhører kun en bruger (Users til Orders) . Hver ordre kan indeholde flere cupcakes, men en Orderdetails hører kun til en ordre (Orders til OrderDetails). Bottoms og toppings kan bruges i flere Orderdetails, men hver linje har kun en Bottom og en Topping (Bottoms → OrderDetails og Toppings → OrderDetails). Dette gør det muligt at genbruge smage og registrere sammensætningen af hver cupcake korrekt.

Nøgler

Alle tabeller bruger auto-genererede ID'er som primærnøgler. Vi anvender også fremmednøgler som b.la.

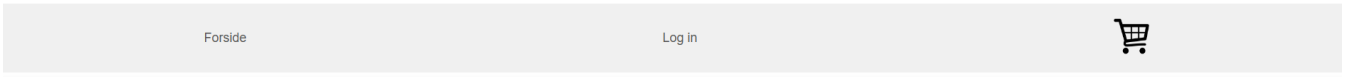
- customer_id i Orders forbinder en ordre til en bruger.
- order_id i OrderDetails sikrer, at detaljer hører til en specifik ordre.
- topping og bottom i OrderDetails linker til de valgte ingredienser.

Navigationsdiagram

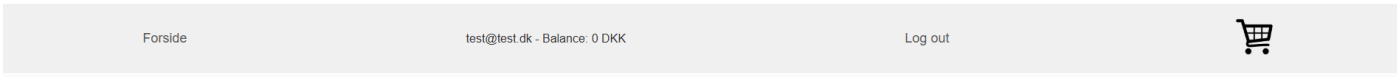
For at give et overblik over vores løsning og dens indhold, har vi lavet et oversigtsdiagram, som kan ses længere nede.

Hjemmesiden indeholder en navigationsbar som er til stede på alle siderne. Oversigtsdiagrammet viser de knapper som navigationsbaren indeholder og de tilsvarende sider som de fører til.
Navigationsbaren har 3 forskellige tilstande som er afhængig af om man er logget ind og om ens bruger er administrator.

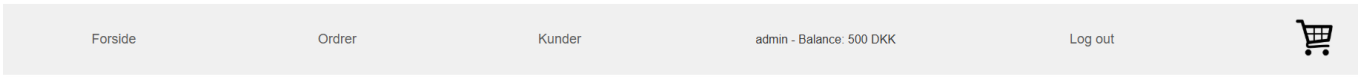
Første tilstand vises som udgangspunkt, når man åbner siden, og ikke er logget ind endnu.



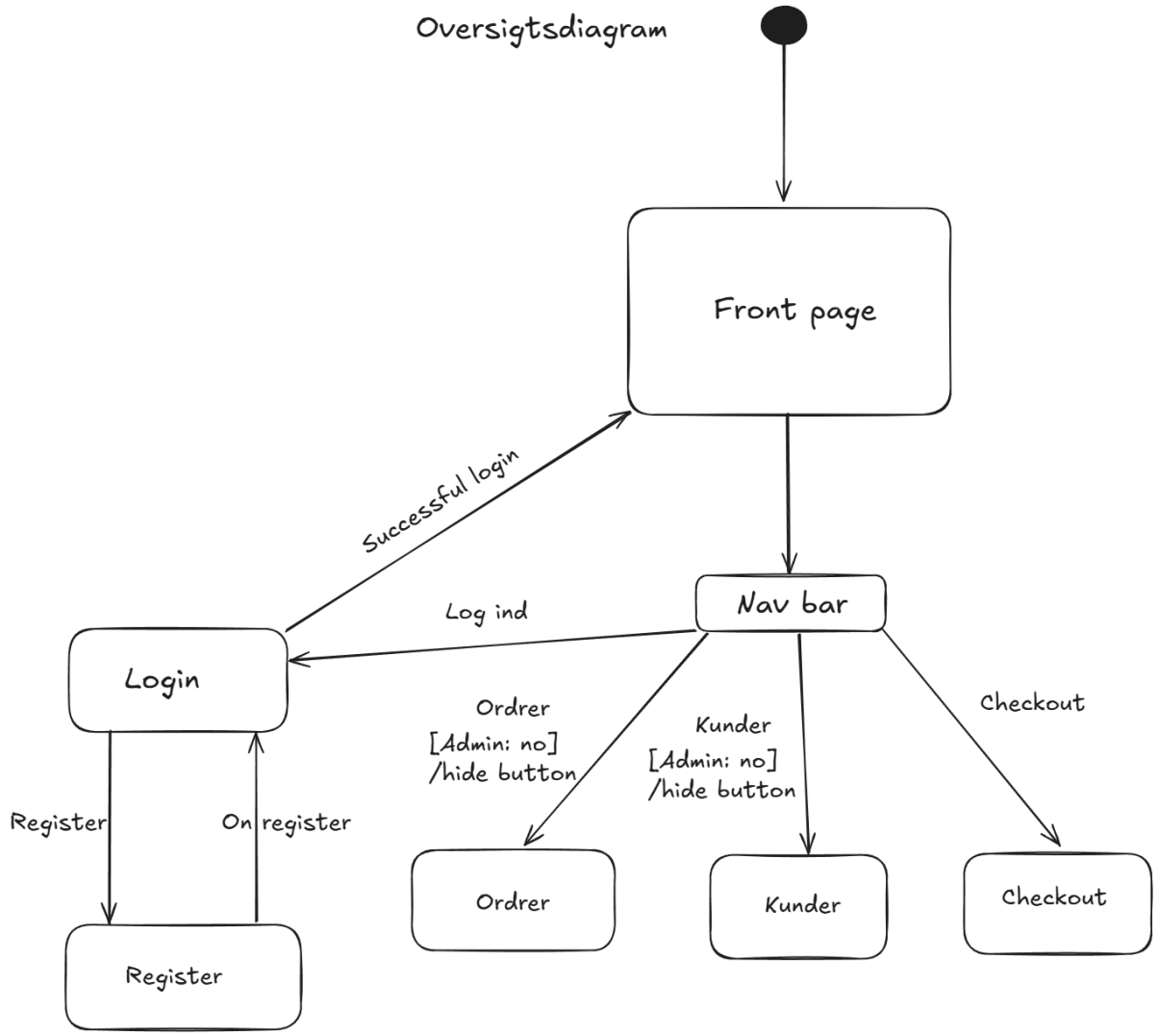
Anden tilstand vises når en bruger er logget ind på siden.



Den tredje tilstand vises hvis brugeren der er logget ind på siden, også er registreret som administrator.



Knapperne “Ordrer” og “Kunder” vises kun hvis det er en admin bruger som er logget ind.



Særlige forhold

- Der gemmes to objekter i session. Når man åbner siden gemmes der et Basket objekt, og når en bruger logger ind gemmes der et User objekt.
- Der gemmes ved tre forskellige scenarier en besked med navnet “error” i session, som bruges til at give en fejlbesked til brugeren.
- Diverse brugerinput på siden valideres i html filerne. Eksempelvis valideres e-mail under registrering ved være sat til type=”email”.

Status på implementation

I vores implementation af Olsker Cupcakes nye webshop har vi nået at style hjemmesiden så den opfylder kravene fra kundens mockup, og alle 9 user-stories er opfyldt og fungerer på hjemmesiden.

De følgende ting mangler fortsat på projektet:

- CRUD metoder til alle tabellerne
- JUnit testing af vores metoder

Proces

Vi valgte som det første i projektforløbet at uddelegere rollerne som blev foreslået fra start af. De blev fordelt således:

- **Scrum-master:** Emil
- **Product-owner:** Markus
- **Tech-lead:** Thor

Vi forsøgte at følge rollerne til at starte med, men efter de første par dage, da vi var kommet godt i gang, smeltede de lidt sammen og vi fokuserede ikke lige så meget på det.

I et fremtidigt projekt syntes vi at det ville være meningsfuldt at prøve at fokusere mere på det, sådan at man fulgte og brugte rollerne mere i projektarbejdet.

Da vi først gik i gang med at kode på projektet, var vi rigtig gode til at arbejdsfordele mellem os alle tre, og møde de deadlines som vi satte.

Vi fik ikke tænkt testing ind i udviklingen af vores projekt, og det blev for stor en opgave at skrive tests da vi først blev opmærksomme på at vi manglede det. Dog fik vi kort før afleveringen udbarjedet en test af vores UserMapper klasse.
Læringen herfra er at i vores næste projekt skal vi tænke tests ind i vores udviklingsproces fra starten af, og forsøge at lave test-driven development.