Cupcake

Deltagere i Gruppe-E:

- Linus Lohmann Mølgaard, cph-lm440@stud.ek.dk, Linus-llm, F25

- Asger Lønstrup Ammitzbøll, cph-aa645@stud.ek.dk, AsgerLA, F25

Tidspunkt for oprindelse:

Rapporten: 28-10-2025
Projektet: 20-10-2025

Indholdsfortegnelse:

Indledning

Dette projekt omhandler at skabe en webshop til firmaet Olsker Cupcakes. Dette har vi gjort vores forsøg på ud fra indsamlede user stories. Vi har bygget det op i etaper af user stories.

Baggrund

Virksomheden er Olsker Cupcakes, og er et iværksættereventyr fra Bornholm. De mener, at de har ramt den rigtige opskrift. Derfor, efter at de har fået lavet et mockup til en forside, er der kommet user stories til for at kunne beskrive systemet.

Kunden kan oprette en profil og dernæst logge ind. Når kunden er logget ind, er der mulighed for at bestille en eller flere cupcakes med valgfri bund og top. Kunden kan vælge at gå videre med ordren og betale. Kunden har også mulighed for at se sine egne ordrer. Der er altid mulighed for at logge ud.

Teknologivalg

Java 17
Maven 3
Database:

- jdbc
- Postgresql 42.7.7
- HikariCP 5.1.0

Web:

- javalin 6.7.0
- javalin-rendering 6.7.0
- thymeleaf 3.2.1.RELEASE

Test:

- junit-jupiter 5.10.2

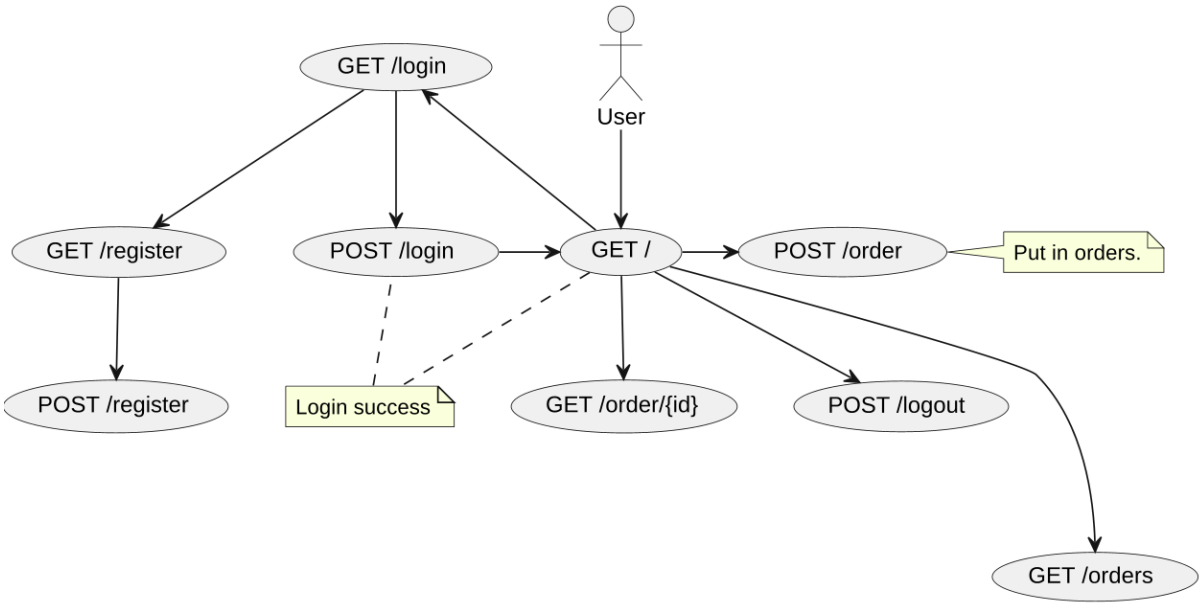
Krav

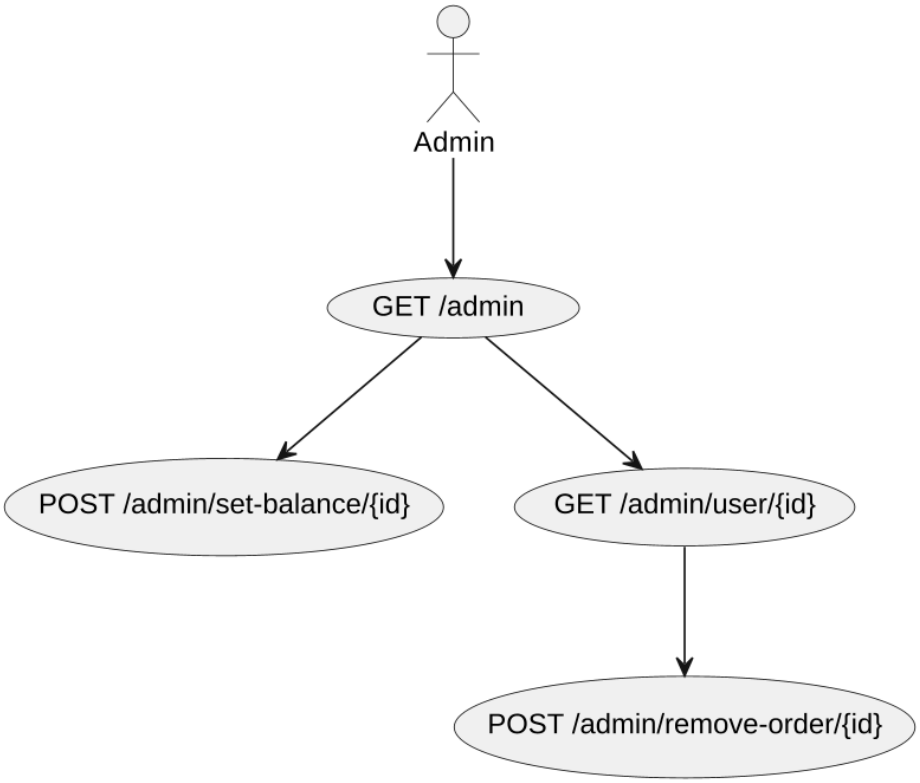
Visionen for systemet og dette projekt i forhold til Olsker Cupcakes, er at de har en fungerende webshop, hvor der kan bestilles cupcakes. Udover det skal der være mulighed for virksomheden, at have en administrator der gør det muligt at skabe overblik over ordrer, kunder og mulighed for at fjerne en ordre. Dette vil skabe en mulighed for at indbringe kunder på en ny måde.

- US-1: Som kunde kan jeg bestille og betale cupcakes med en valgfri bund og top, så 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 e-mail 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, så 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.

Use case diagram

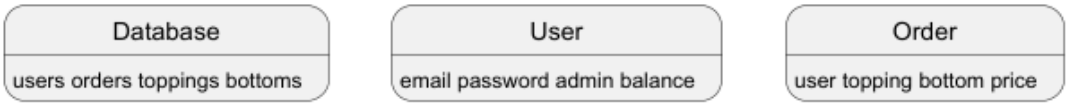



Admin
GET /admin
POST /admin/set-balance/{id}
GET /admin/user/{id}
POST /admin/remove-order/{id}

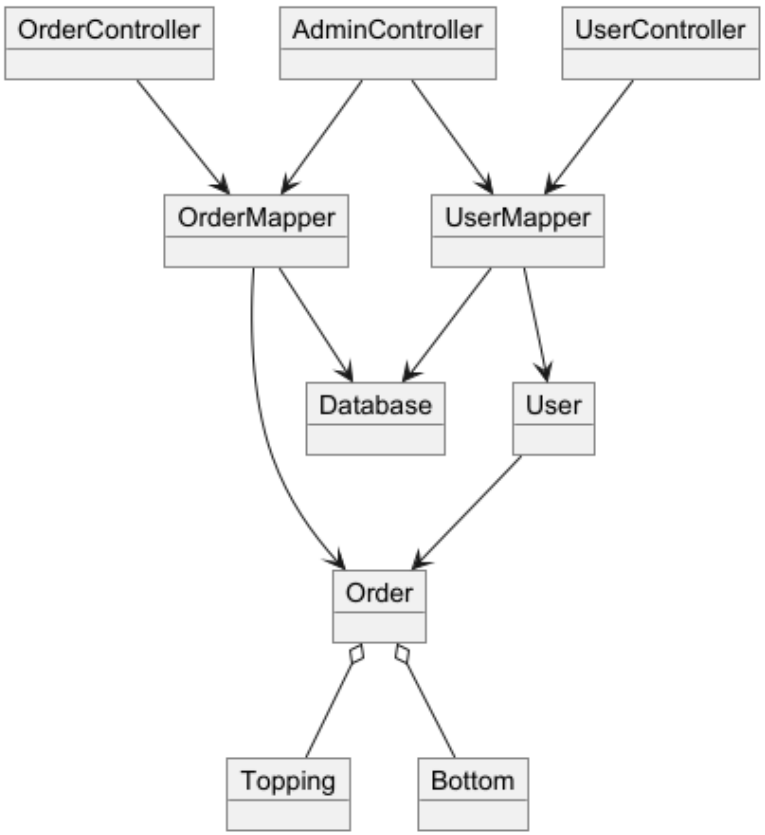
Domæne model og ER diagram

De største overvejelser i forbindelse med vores domænemodel var, hvordan Order skulle sættes sammen. Vi overvejede lidt om topping og bottom skulle være attributter inde i orders samlet, men vi endte på at der er to strings inde i Order. De to strings holder nu navnet på topping'en og bottom'en.

Dette er den første domænemodel, vi udarbejdede:

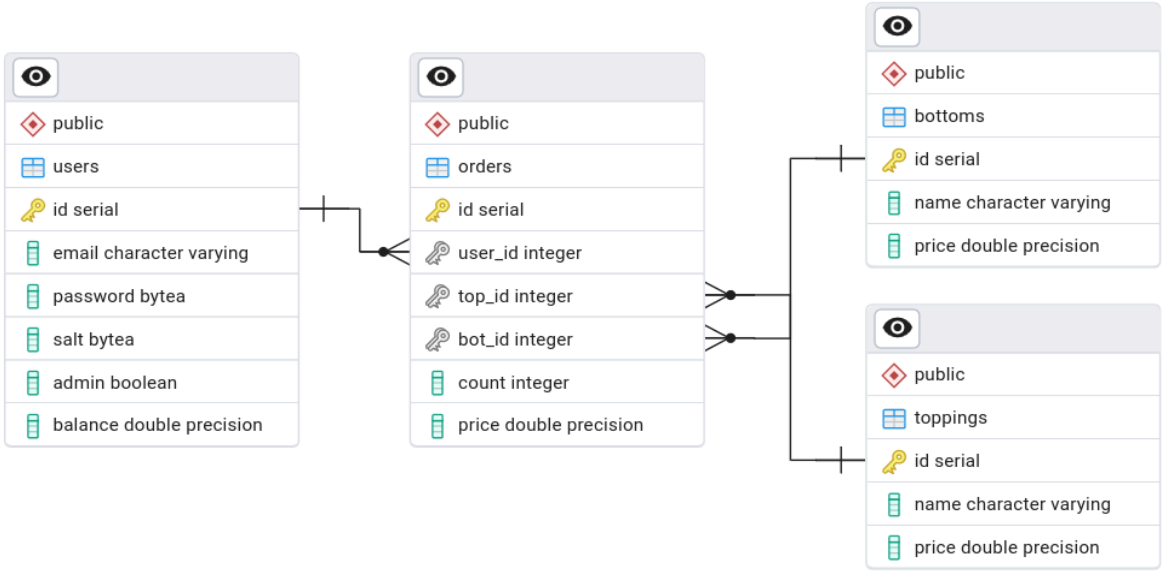


Dette er den nye domænemodel:

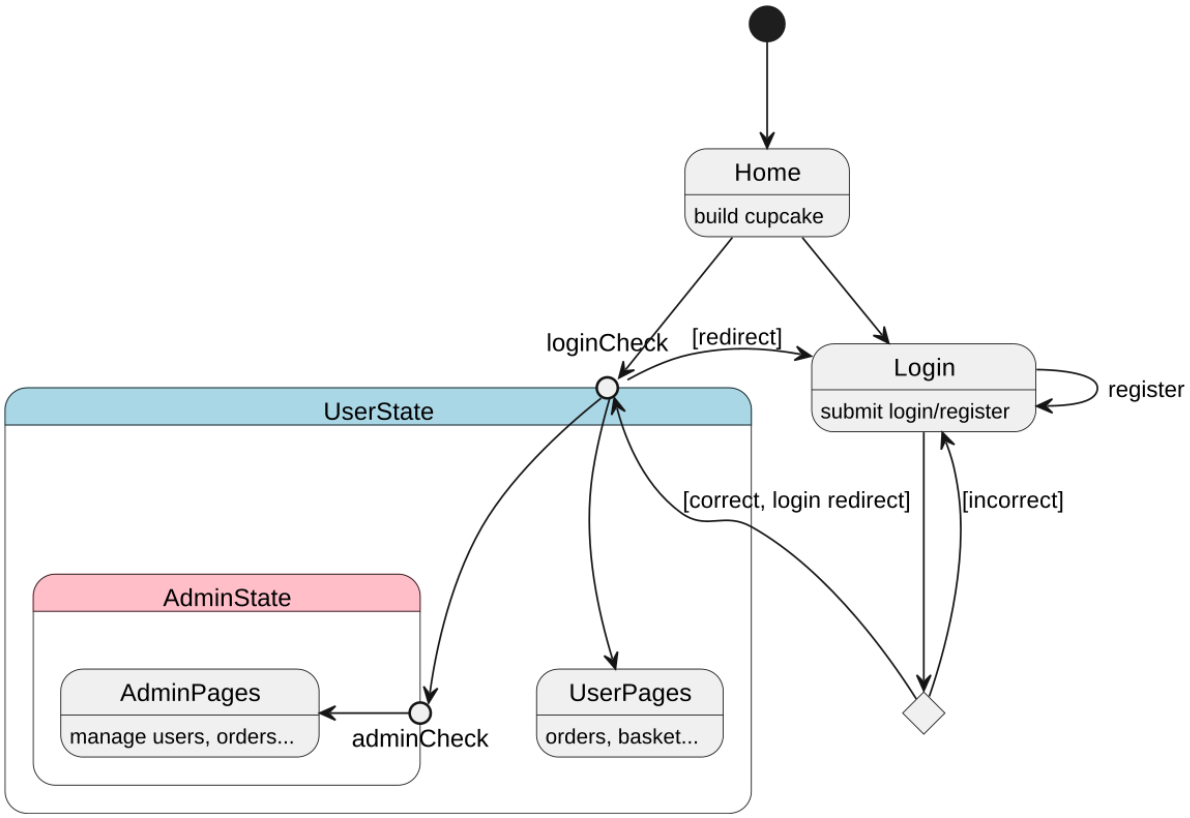


ERD:

Det kom rimelig hurtigt til os at orders bliver samlingspunktet for alle, så derfor har den tre fremmednøgler. Kolonnen price i orders overholder ikke tredje normalform, da den afhænger af de to price kolonner i bottoms og toppings. Dog overholder den anden normal form. Vi tænkte, at siden at prisen ikke ændrer sig i orders selvom, at man ændrer prisen i toppings eller bottoms, så har det ikke den største betydning at den ikke overholder tredje normalform.
Vi ville gerne prøve det af med kryptering af password, så det har vi taget med.



Navigationsdiagram



Forsimplet udgave grundet overblik.

Særlige forhold

session attributes:

- User “user” - sat når user er logget ind
- String “errmsg” - fejlbeskeder
- String “loginredirect” - redirect efter login
- List<Order> “basket” - ordrer i indkøbskurven
- double “subtotal” - samlet pris i indkøbskurven

Exceptions:

Exceptions der ikke bliver håndteret i Controllers, sender en “500 Server Error” side. Post handlers fanger exceptions og sender “400 Bad Request”.

Login:

Hvis sessionAtrribute(“user) == null, så er user ikke logget ind.
Alle routes til “/admin*” bliver redirected til “/login”, hvis user == null eller user ikke er admin. Kodeordet bliver hashed med algoritmen “PBKDF2/HMAC/SHA1” implementeret i javax.crypto package.

Status på implementation

- Vi har ikke lavet aktivitetsdiagram, men vi har lavet use case diagram i stedet.
- https://youtu.be/2JVDahiEAaE (video link til fremvisning)

Proces

Vores plan var at tage det løst, så hvis en af os begyndte på noget, begyndte den anden på noget andet. Der har ikke været nogle regler i forhold til arbejdstidspunkter eller hvor meget tid i alt. Vi har talt om hvad der skal laves, og så har vi lavet så meget vi kunne, indtil vi følte at det var passende for dagen.
Vi synes at det har fungeret godt, og har ikke umiddelbart noget der kunne gøres bedre. Det er meget rart at kunne være fleksibel med tidspunkt og arbejde, da giver noget frihed.