

Olivia Sønder Larsen- cph-ol75@stud.ek.dk1

Hussain Lafteh- cph-ha274@stud.ek.dk

EK- Datamatiker, hold Efterår 2025, sem 2

30/10-2025



Cupcake projekt

Af Hussain og Olivia

Indholdsfortegnelse

1. Forside.....	s. 1
2. Indholdsfortegnelse.....	s. 2
3. Indledning.....	s. 3-4
4. Krav.....	s. 5-6
5. Aktivitetsdiagram.....	s. 7
6. Domænemodel og ERD.....	s. 8-9
7. Navigation diagram.....	s. 10
8. Særlige forhold.....	s. 11
9. Status på implementation.....	s. 12-13
10. Process.....	s. 14

Indledning

Denne rapport beskriver udviklingen af et websystem kaldet *Olsker Cupcakes*, som er et fiktivt kundeprojekt udarbejdet som en del af Datamatikeruddannelsens 2. semesters projektforløb. Formålet med projektet er at udvikle en fungerende webshop, hvor kunder kan sammensætte deres egne cupcakes med valgfri bund og topping, se deres ordre, oprette en profil og gennemføre et køb.

Projektet tager udgangspunkt i en opgave stillet af *Olsker Cupcakes* — en fiktiv, bæredygtig virksomhed fra Bornholm, som ønsker en digital løsning til at håndtere kunders cupcakebestillinger. Der er udleveret en løs mockup af en forside, samt en række user stories, der beskriver de vigtigste funktionelle krav til systemet. På baggrund af disse krav er der udviklet en prototype, som gradvist udbygges med funktionalitet såsom login, ordreoversigt og databaseintegration.

Projektet er udviklet i **Java 17** ved brug af **Javalin** som webframework, **Thymeleaf** som template engine og **PostgreSQL** som database. Derudover anvendes **HTML** og **CSS** til struktur og styling af webapplikationen.

Alt kildekode versionsstyres via **Git** og deles på GitHub, og rapporten ledsages af en kort videodemonstration af det færdige system.

Formålet med rapporten er at dokumentere projektets udviklingsproces, designovervejelser og tekniske implementering, så en fagfælle kan forstå, reproducere og vurdere løsningen ud fra et både teknisk og metodisk perspektiv.

Baggrund

Virksomhedsbaggrund:

Olsker Cupcakes, et dybdeøkologisk iværksættereventyr fra Bornholm, som har ramt den helt rigtige opskrift. Virksomheden ønsker en simple hjemmeside til at sælge deres cupcakes.

Virksomhedens ønsker til løsningen:

Kunden ønsker et system, hvor brugeren kan sammensætte og bestille cupcakes ved at vælge både en bund og en topping. Det skal være muligt at se en oversigt over sine valgte cupcakes i en indkøbskurv, og herefter kunne afgive en ordre.

Olivia Sønder Larsen- cph-ol75@stud.ek.dk4
Hussain Lafteh- cph-ha274@stud.ek.dk
EK- Datamatiker, hold Efterår 2025, sem 2
30/10-2025

Systemet skal desuden give mulighed for, at kunder kan oprette en konto og logge ind, så de kan gemme og betale for deres ordrer. Administratoren skal kunne se alle kunder og ordrer, samt kunne administrere dem i systemet.

Yderligere må systemet godt kunne gemme data om ordrer, kunder, bunde og toppings i en database, så informationen ikke går tabt. Løsningen skal være brugervenlig og fungere som en simpel, digital webshop for Olsker Cupcakes.

Teknologi valg

Til denne Opgave har vi brugt følgende teknologier:

- IntelliJ IDEA 2024.3.4.1
- Java v. 17
- Postgres
- Javalin
- Thymeleaf
- JDBC
- HTML
- CSS
- JavaScript

Krav

Formålet med Olsker Cupcakes' online bestillingssystem

er at give kunderne en nem og fleksibel måde at bestille cupcakes på, samtidig med at virksomheden kan optimere drift og salg. Systemet skal:

1. Gøre det muligt for kunderne at bestille cupcakes online fra computer, tablet eller mobil – uanset hvor de befinder sig.
2. Minimere ventetid ved at lade kunderne vælge afhentningstidspunkt eller levering direkte gennem systemet.
3. Øge salget ved at nå ud til nye kunder, som ellers ikke besøger bageriet fysisk.
4. Forbedre lagerstyringen ved automatisk at registrere bestillinger og justere produktionen, så spild reduceres.
5. Indsamle data om kundernes smagspræferencer og købsmønstre, så menuen og kampagner kan tilpasses efter efterspørgslen.

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 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.

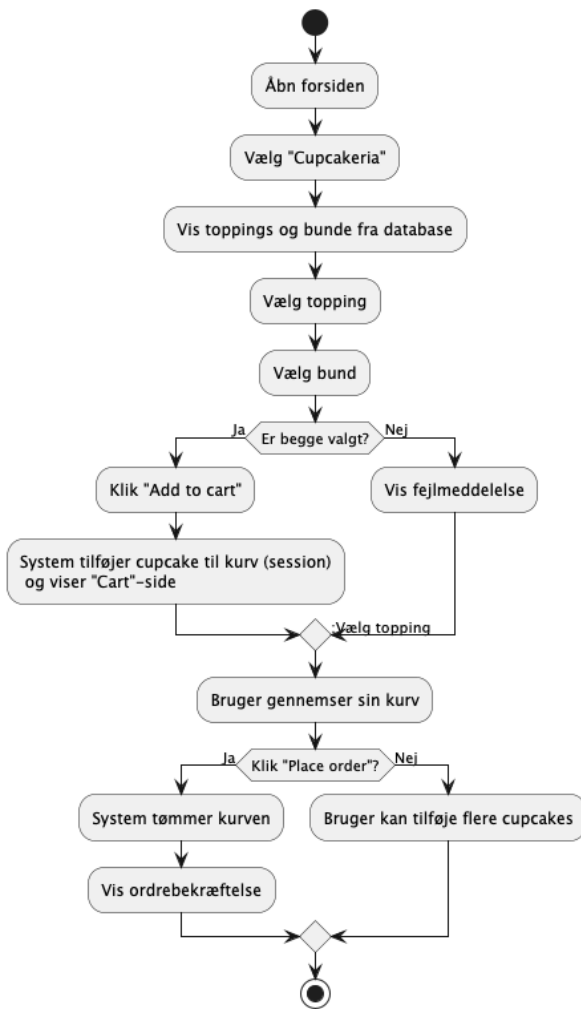
US-7: Som administrator kan jeg se alle kunder i systemet og deres ordrer, sådan at jeg kan følge op på ordrer og holde styr på mine kunder.

Olivia Sønder Larsen- cph-ol75@stud.ek.dk6
Hussain Lafteh- cph-ha274@stud.ek.dk
EK- Datamatiker, hold Efterår 2025, sem 2
30/10-2025

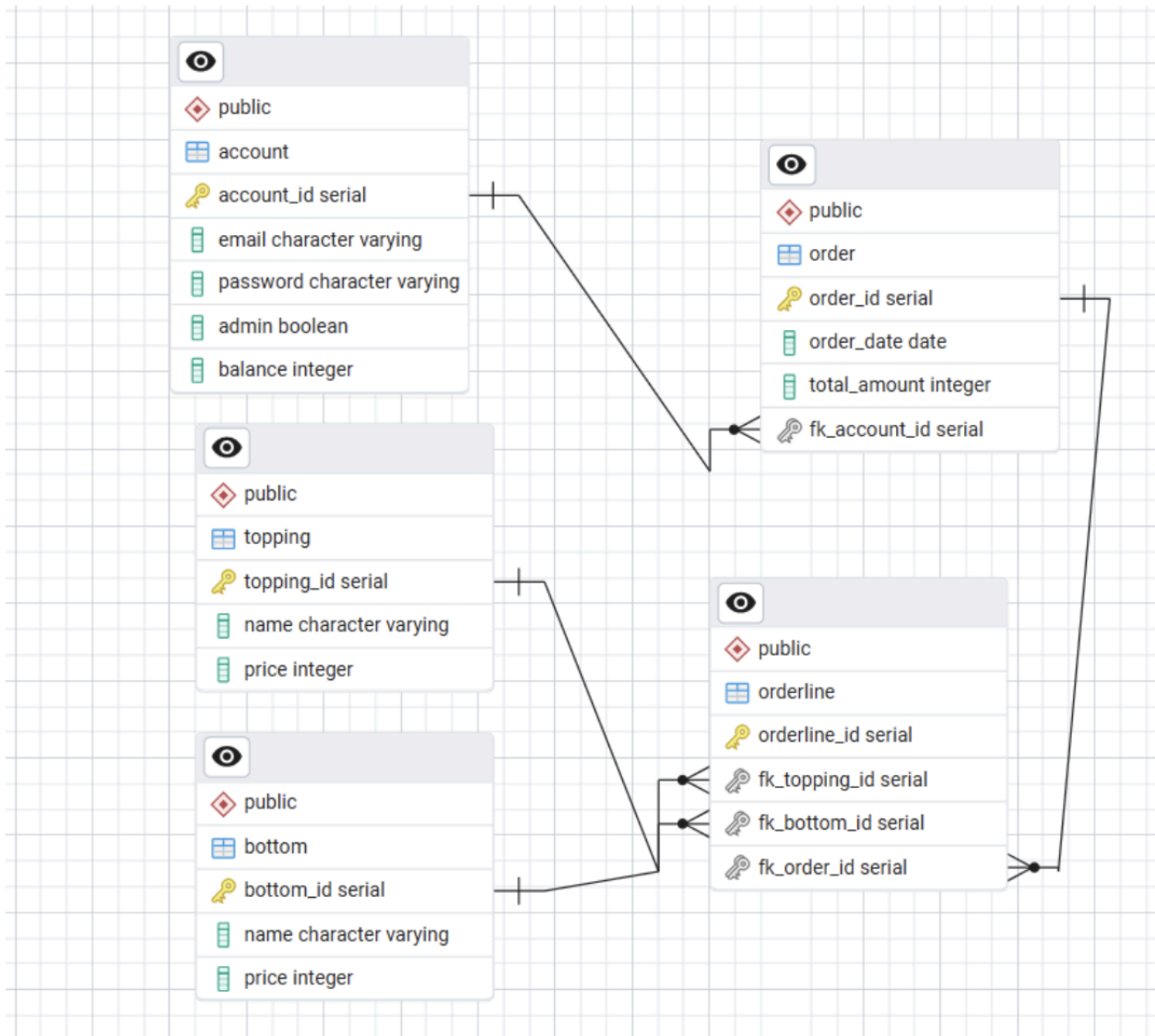
US-8: Som kunde kan jeg fjerne en ordrelinie fra min indkøbskurv, så jeg kan justere min ordre.

US-9: Som administrator kan jeg fjerne en ordre, så systemet ikke kommer til at indeholde ugyldige ordrer. F.eks. hvis kunden aldrig har betalt.

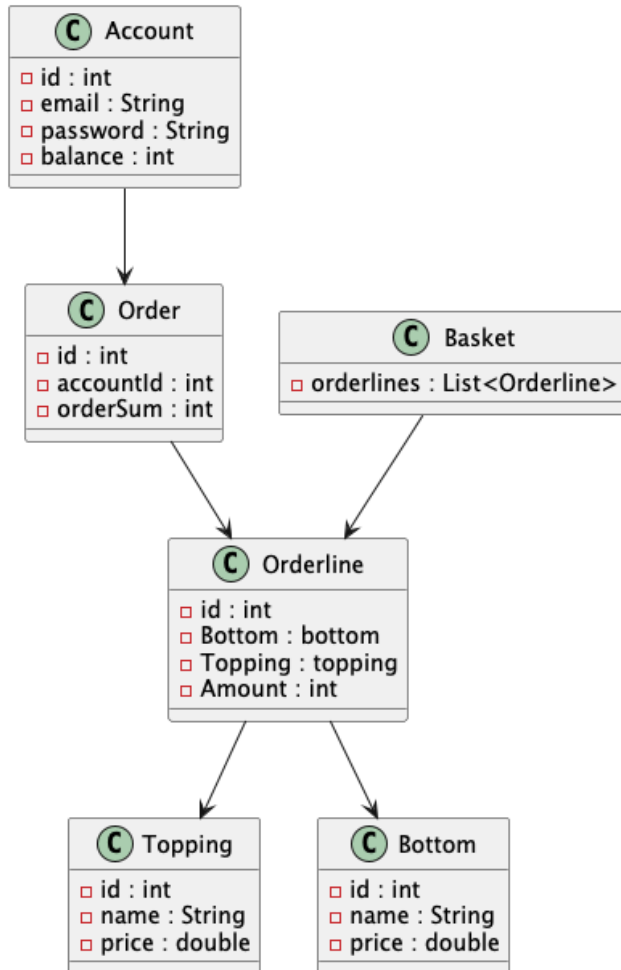
Aktivitetsdiagram



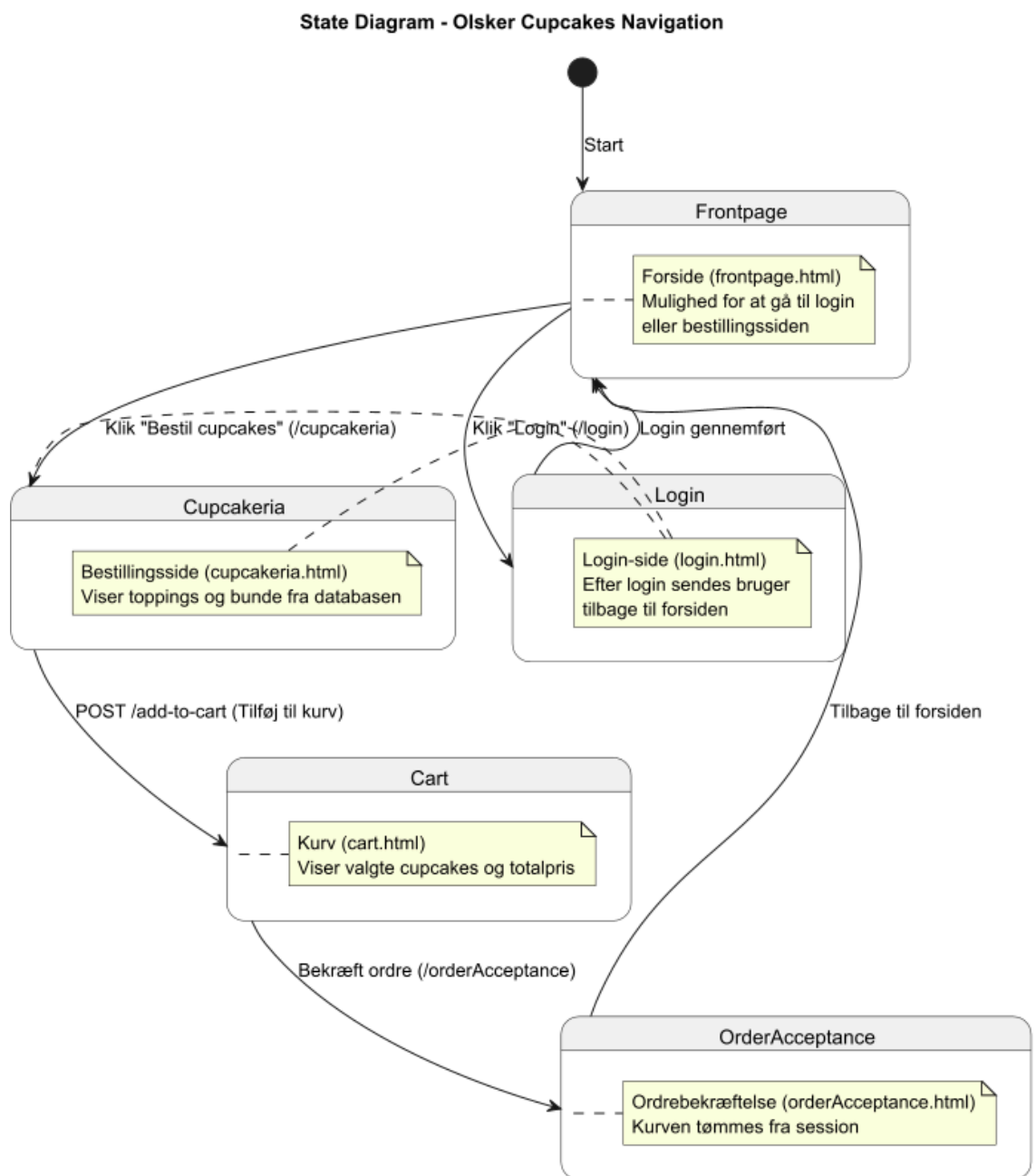
Domæne model og ER diagram



Domænemodel(til visning af forretningslogik)



Navigationsdiagram



Særlige forhold

Brug- og valg af entity-constructors i controller

I vores entities har vi lavet flere forskellige constructorer, såsom i vores orderline-entity, hvor at vi har lavet hhv. En constructor til admin-brugere og normale brugere. I vores orderController har vi valgt at starte med at lave metoder der tager udgangspunkt i constructoren til normale brugere.

Brug af JavaScript og Thymeleaf til valg af cupcake, i form af gliders

Vi har valgt at gøre lidt mere ud af vores cupcake-vælger, ved at erstatte vores dropdowns med visuelle gallerier/gliders af hver topping og bund.

Til at udføre dette har vi brugt JavaScript og importeret et JavaScript-bibliotek kaldet [Glider.js](#), som tilbyder et template til at lave dynamiske billede-karouseller.

Senere fandt vi ud af at det mest oplagte, for yderligere at tilføje værdier og funktion til disse billede-karousellerer var at bruge Thymeleaf, hvorfor at vores backend er blevet en kombination af Javalin, Thymeleaf og JavaScript. Javalin håndterer data, Thymeleaf integrerer data i HTML'en, og JavaScript står for den visuelle og interaktive del.

Status på implementation

Opgave	Status	Uddybelse
US - 1	Done	
US - 2	Done	
US - 3	Done	
US - 4	Done	
US - 5	Partially done	Der kan oprettes en bruger og logges ind, men har en mangel der gør at du ryger til en blank side, når man logger ind.
US - 6	Done	
US - 7	Done	
US - 8	Not done	
US - 9	Not done	
Styling	Done	
Figma mockup	Partially done	Vi glemte først at lave et mockup og gik direkte til html og styling, med inspiration fra pinterest og allerede eksisterende, bageriers hjemmesider.

Mangler/bugs:

- Fejl ved gliders: Når man når til det sidste element i gliders, kan man ikke gå tilbage igen, med prev-button.
- Fejl ved headeren: Når man står på order acceptance-siden, kan man ikke gå til cart-siden, fra header-menuen.
- Frontend-layout: Vi ville gerne have finjusteret front-end-layoutet endnu mere, men glemte det til sidst om fik ikke tid til det.

Olivia Sønder Larsen- cph-ol75@stud.ek.dk13

Hussain Lafteh- cph-ha274@stud.ek.dk

EK- Datamatiker, hold Efterår 2025, sem 2

30/10-2025

- Mangel på siden efter login: Når man har logget ind kommer man til en blank side.

Process

Olivia Sønder Larsen- cph-ol75@stud.ek.dk14
Hussain Lafteh- cph-ha274@stud.ek.dk
EK- Datamatiker, hold Efterår 2025, sem 2
30/10-2025

I starten af projektet planlagde vi at arbejde som et team, hvor vi mødtes jævnligt for at fordele opgaver og sikre, at alle var opdateret. Vores plan var at samarbejde tæt, men samtidig arbejde selvstændigt med hver vores ansvarsområde mellem møderne. Vi besluttede at mødes cirka hver anden eller tredje dag for at gennemgå fremskridt, uddele nye opgaver og hjælpe hinanden med de udfordringer, der opstod.

Vi fordelte rollerne, så Olivia havde ansvaret for frontend-delen, hvor hun stod for design, struktur og opsætning af HTML-siderne, CSS og thymeleaf-funktionalitet, imens Hussain arbejdede med databaselaget, mappers, entities og database-connection.

Backend-funktionaliteten, såsom controllers, arbejdede vi sammen om at bygge. Med hver vores ekspertise og kendskab til programmet gav det god mening at løse sammen.

I praksis gik samarbejdet overordnet godt. Vi havde en god kommunikation, og begge bidrog aktivt. Dog opstod der en udfordring i forhold til rækkefølgen af arbejdet. Vi startede nemlig med at lave frontend først og forsøgte derefter at få backend til at passe ind bagefter. Det viste sig at være mere kompliceret end forventet, fordi strukturen i frontend ikke helt matchede den logik, som backend krævede. Det betød, at vi måtte ændre og tilpasse en del kode for at få de to dele til at fungere sammen.

Det, der fungerede godt, var samarbejdet og fordelingen af ansvar – vi havde klare roller, og kommunikationen under møderne var effektiv. Det, der kunne have været bedre, var vores planlægning af rækkefølgen i udviklingen. Næste gang vil vi starte med at få backend og databaseforbindelserne på plads først og derefter bygge frontend ovenpå det.

Vi har lært, at det er vigtigt at tænke hele systemets struktur igennem fra starten og at bygge i lag, så backend og frontend udvikles i takt med hinanden. Derudover har vi fået erfaring med at samarbejde mere effektivt i et lille team, fordele ansvar og løse problemer sammen undervejs.