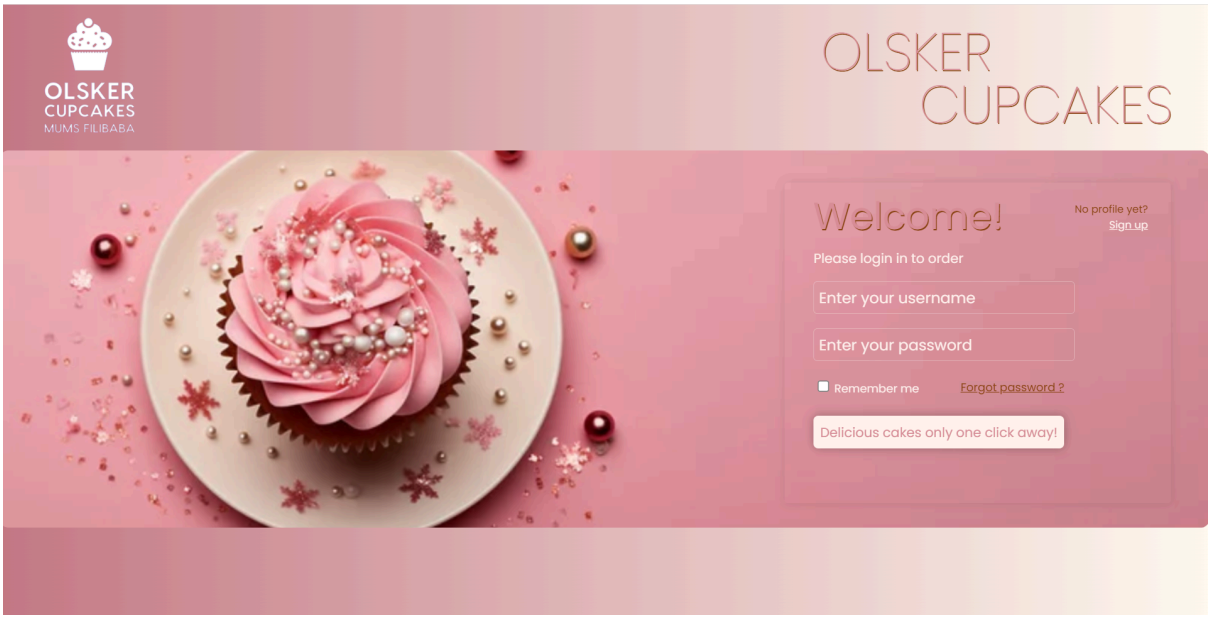
Cupcake projektrapport



Navn: Ali El-Seelawi **Mail:** cph-ae183@cphbusiness.dk **Git:** Alibabaaa444 **Hold:** A	**Navn:** Fatima Majid Shamcizadh **Mail:** Cph-fs156@cphbusiness.dk **Git:** Fati01600 **Hold:** A	**Navn:** Juvena Akua Walters **Mail:** cph-jh662@cphbusiness.dk **Git:** Juvenaawh **Hold:** A

Projektet blev udarbejdet: 01.04.2024-11.04.2024

Indholdsfortegnelse

Indledning

Denne indledning er møntet på og formuleret til en fagfælle, altså en datamatiker studerende på 2. semester, der er på samme niveau som gruppemedlemmerne.

Vi skulle udvikle en hjemmeside til et bageri, så deres kunder fremadrettet også kunne bestille cupcakes online. Der skulle både laves backend- og frontend til dette projekt. Det skulle være en dynamisk hjemmeside som skulle kunne håndtere kontoer- og bestillinger, og systemet blev udviklet i IntelliJ og programmeret i Java.

Vi startede med at lave et mockup af hjemmesiden i Figma for at få en ide om layoutet. Med PostgreSQL lavede vi et diagram, der viser databasens struktur. Efter oprettelsen af databasen kunne kunderne oprette en konto og bestille kager, som bageriet derefter kunne forberede til afhentning. Forbindelsen til databasen blev håndteret med HikariCP, hvilket gør det muligt for hjemmesiden at hente og vise information om kunder og deres ordrer.

Baggrund

Olsker Cupcakes, som er en virksomhed placeret på Bornholm, har et ønske om at få lavet en hjemmeside, der har til formål at digitalisere kundeoplevelsen og gøre det nemmere for kunderne at bestille lige netop den cupcake, der vækker den søde tand, ligeledes at kunderne kan vælge at personliggøre deres cupcake, ved selv at vælge bunden og topping. Vi startede med at få en grundlæggende forståelse for, hvad de havde brug for, ved at se på hvad hipsterne fra København havde skitseret. De skitserede en forside og samlede nogle krav, som vi fik inspiration af og byggede videre på. Dette gjorde vi ved at analysere manglende funktionalitet, stille spørgsmål og komme med forslag til forbedringer.

Teknologivalg

Vi valgte nogle kendte og stabile teknologier, så andre nemt kan arbejde videre på projektet efter os:

IntelliJ IDEA 2023.2.2 (Ultimate Edition)	Dette er vores udviklingsprogram.
Java (17 Amazon Coretto)	Vi brugte programmeringssproget Java til backend delen.
Javalin (6.1.3)	Web-framework, brugte vi til at oprette webserveren.
Thymeleaf (3.1.2 RELEASE)	Til at indsætte server- side data i vores html skabelon.
HikariCP (5.1.0)	For at håndtere database forbindelser via connection pooling, effektivt.
Postgresql (42.7.2)	Som vores database til at lagre dataen.
Github (3.12.1)	Brugt til at samarbejde med hinanden ved at

	pushe og committe og lave Kanban board, så vi hele tiden havde et overblik over hvem der laver hvad og hvor langt man var i processen.

Krav

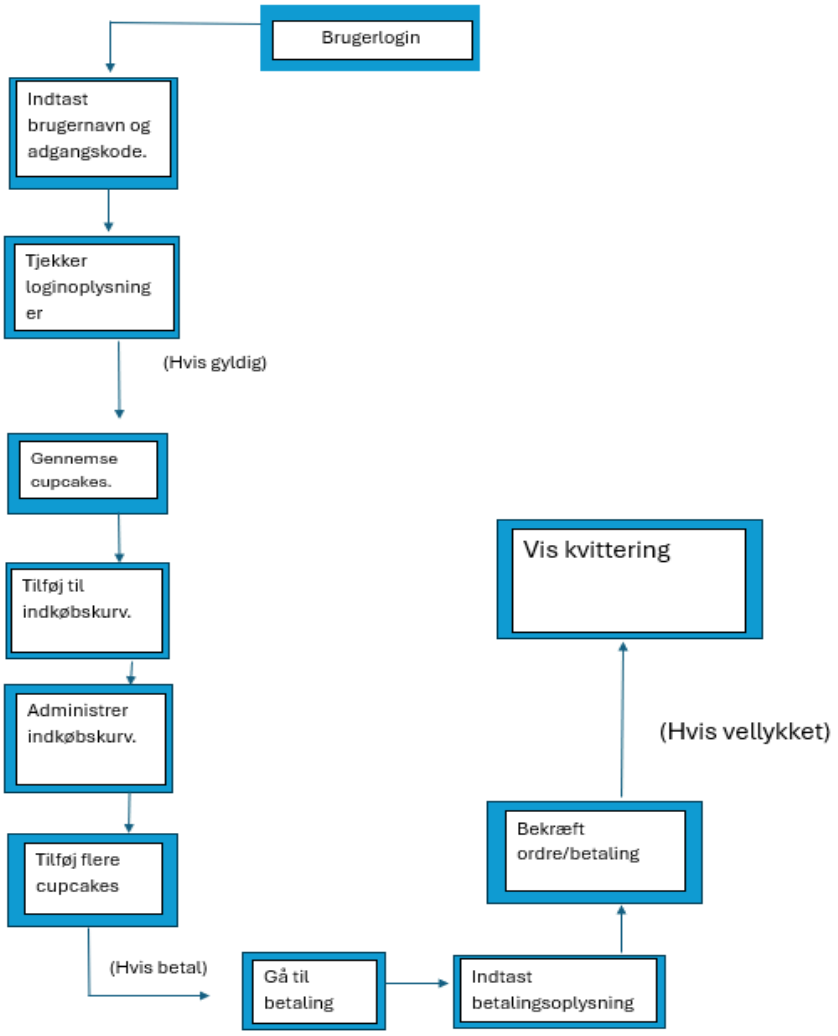
Virksomhedens vision er at lokke flere kunder til at købe cupcakes hos Olsker Cupcakes, da det nu er gjort nemmere tilgængeligt og mere belejligt for kunden ved, at de kan sidde i ro og mag i sofaen og vælge lige præcis de cupcakes de ønsker, til hvilken som helst lejlighed.

Første kundemøde mundede ud i disse user-stories:

- US-1: Kunder skal kunne bestille og betale for cupcakes med det bund og topping de vil have.
- US-2: Kunder skal kunne lave en konto for at gemme deres bestillinger.
- US-3 til US-9: Der er også mål for hvordan vi som laver hjemmesiden, skal kunne se og håndtere ordrerne."

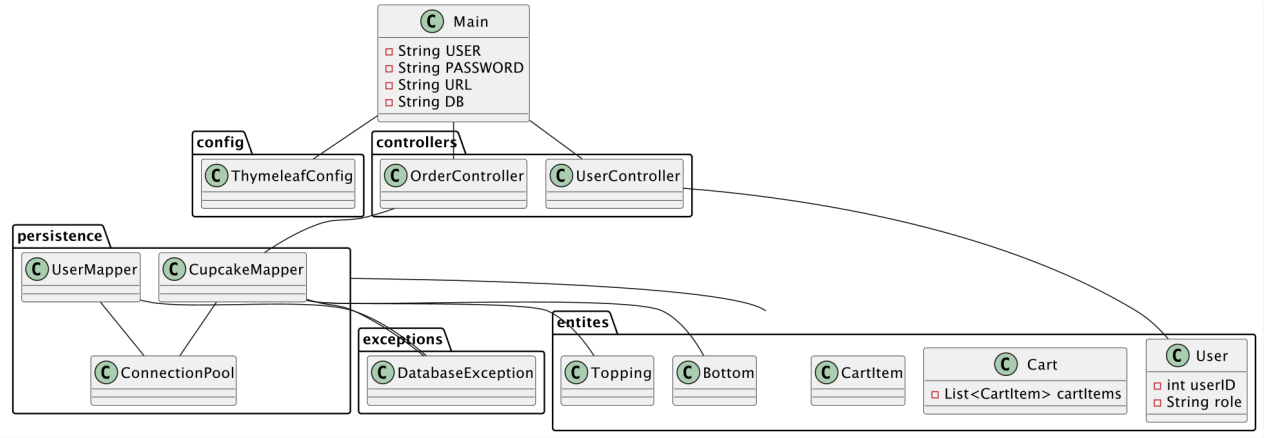
Aktivitetsdiagram

Kunder starter med at logge ind. Hvis de skriver det rigtige kontonavn og kodeord, kan du se cupcakes og lægge dem i deres indkøbskurv. De kan betale for deres cupcakes online, og hvis betalingen går igennem, får de en kvittering. Hvis der er problemer med betalingen, skal de prøve at betale igen.



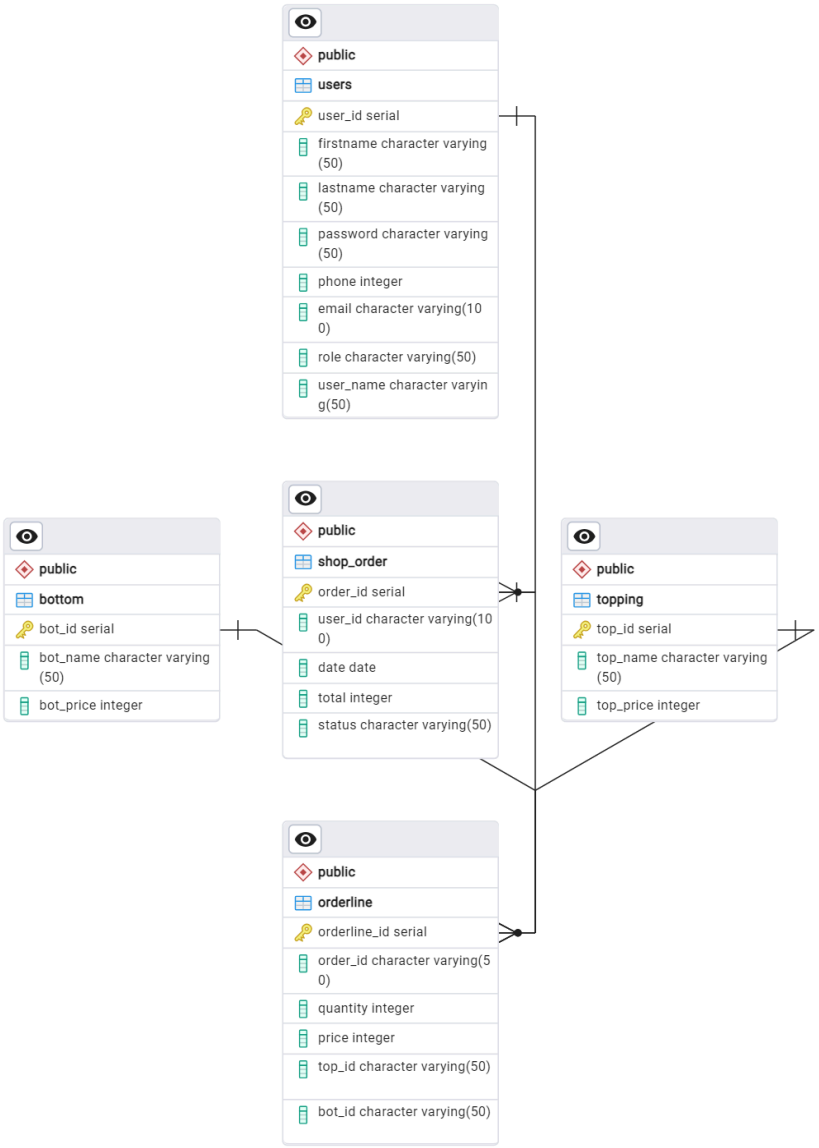
Domænemodel

Systemet er designet til at håndtere kontoer, produkter og ordrer. Det forbindes til en database via en ConnectionPool, som tillader det at hente oplysninger om kundernes konti og de produkter bageriet sælger. User- og CupcakeMapper klasserne trækker denne information fra databasens tabeller. User- og OrderController klasserne bruger metoder til at sende og modtage data via POST og GET requests til Javalin serveren. ThymeleafConfig sørger for at indlæse denne data i HTML gennem session objekter for en dynamisk kundeoplevelse.



Vi oplevede udfordringer med at skabe en fyldestgørende domænemodel, hvilket begrænsede detaljeringsgraden af vores model. Det her er det bedste vi kunne levere grundet disse komplikationer.

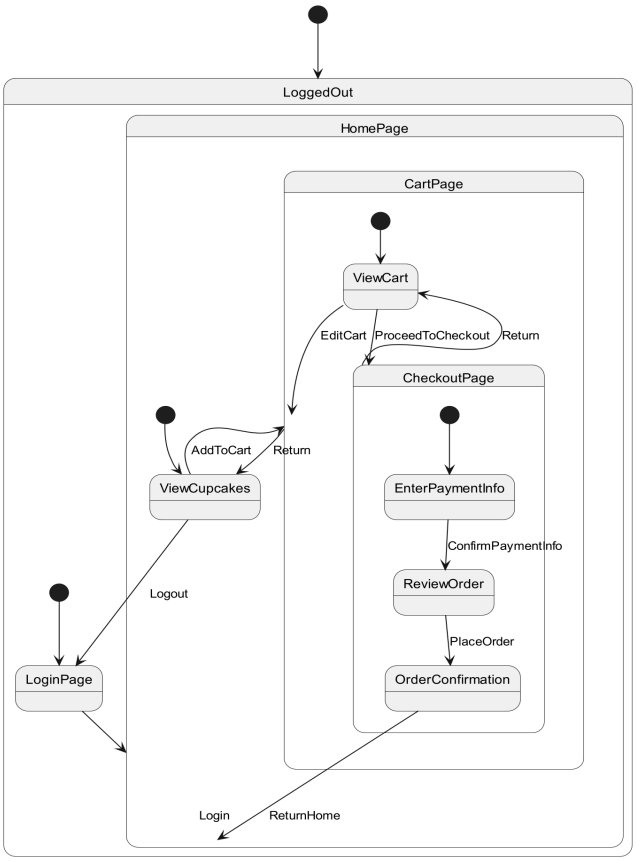
ER diagram



ER-diagrammet skulle være simpelt, konsistent og opfylde de krav der er til et sådant diagram. Kunder skulle kunne oprette en konto ved at indtaste personlige oplysninger og lave et kodeord. Der blev lavet en tabel til både bunde og topping, så kunder selv kunne personliggøre deres cupcakes med bund og topping, og databasen senere hen kunne udvides med flere produkter. I kurven skulle man som kunde kunne se hvilke cupcakes man havde valgt samt mængden af disse og det samlede beløb af ordren. Der blev knyttet et order_id som fremmednøgle til alle ordrelinier, så man efterfølgende vil kunne kigge på sine tidligere køb, og se indholdet, dato og status på disse.

Navigation Diagram

Målet med vores hjemmeside var at gøre den brugervenlig og nem at navigere. Kunden starter på loginskærmen, hvor de skal indtaste loginoplysninger. Hvis disse er korrekte, kommer de ind på hovedsiden, hvor de kan se og tilføje cupcakes til deres indkøbskurv, eller vælge at logge ud. Fra indkøbskurven kan kunden gå til betalingssiden eller vende tilbage til hovedsiden. På betalingssiden indtaster kunden betalingsoplysninger og betaler. Efter bekræftelsen vises en bekræftelsesside med ordredetaljer og mulighed for at vende tilbage til hovedsiden eller logge ud. Dette flow sikrer, at kunden nemt kan finde og købe produkter, samt ændre deres bestillinger som nødvendigt.



Særlige forhold

Vi har implementeret session objekter til at gemme kundens login-status og indholdet af deres indkøbskurv for at bevare kontinuitet under navigationen på hjemmesiden. Når der opstår fejl, som for eksempel databasefejl, vises fejl-beskeder til kunden med det formål at fejlene skal løses hurtigt og effektivt.

Desuden er der i databasen mulighed for at oprette enten normale kunde konti eller administrator konti. Dog blev funktionaliteten, der skulle give administratorer udvidede rettigheder, ikke færdigimplementeret.

Status på implementation

Vi nåede kun at implementere US-1, US-2 (halvt), US-4 og US-5.

Vi fik implementeret en startside med login, en registreringsside, en bestillingsside, en kurv-side og en check-outside.

Systemet burde klart have flere funktioner og kunne mere, men vi var desværre rigtig presset, så vi valgte at fokusere på det vi var i gang med og færdiggøre så meget af det som muligt.

Vi lavede en kolonne i user tabellen som hed role, så man kunne differentiere mellem almindelige konti og administratorer, men dette får vi aldrig gjort brug af.

Man kan som kunde ikke ændre på indholdet i sin kurv eller se sine ordrer efter de er bestilt.

På bestillingssiden kan man af underlige grunde ikke se kurven i menuen, som man kan på de andre sider.

Vi fandt ud af lige inden projektet skulle afleveres, at når man som kunde tilføjede mere end 3 ordrer i sin kurv, var det ikke muligt at “keep shopping” eller “checkout”, da disse knapper så forsvandt.

Proces

Vi indledte projektet med et møde for at diskutere tilgang og opdeling via Kanban, hvilket hjalp os med at strukturere opgaverne og spore fremskridt individuelt. Opgaverne blev fordelt således, at en person tog sig af databasen, en anden af mockup, og en tredje stod for opsætning i IntelliJ og GitHub repository. Dette system fungerede godt i begyndelsen, specielt med de mindre krævende opgaver, da det gav fleksibilitet i arbejdstiderne.

Imidlertid kom privatliv i vejen, hvilket skabte stress og lange arbejdsdage, især mod slutningen af projektet og under rapportskrivningen. De mange individuelle dele, der skulle integreres, gjorde det også udfordrende, og et tidligt problem med et baggrundsbillede i HTML tog uforholdsmæssigt meget tid, hvilket vi burde have omgået tidligere.

Erfaringerne har vist, at flere hænder gør arbejdet lettere, så til fremtidige projekter ville det være fordelagtigt at være flere i gruppen, hvilket kunne reducere stress og arbejdsbyrden betydeligt.