

CUPCAKE PROJEKT



Dino Saldic – cph-ds303@stud.ek.dk – cph-ds303

Oliver Møller-Jensen – cph-om96@stud.ek.dk - olivermjmj

Carl Emil Køhn – cph-ck381@stud.ek.dk – CarlEmilKoehn

DAT F25

Indholdsfortegnelse

CUPCAKE PROJEKT	1
INDLEDNING.....	3
BAGGRUND	3
KRAV	4
Database	5
Teknologier.....	5
<i>Særlige forhold</i>	<i>6</i>
1. Hvad gemmes i sessionen?.....	6
2. Hvordan blev exceptions håndteret?	6
3. Validering af brugerinput:.....	6
4. Hashing af password:	6
5. Brugertyper:	6
<i>Status på implementering</i>	<i>7</i>
<i>Proces.....</i>	<i>7</i>
BILAG	8
<i>Bilag 1 Olsker cupcakes egen mockup af hjemmeside</i>	<i>8</i>
<i>Bilag 2. ERD</i>	<i>9</i>

INDLEDNING

I cupcake-projektet, skulle der ageres som udvikler til Olsker Cupcakes, en lokal bornholmsk virksomhed, der ønskede en hjemmeside til deres cupcake-forretning. Formålet var at udvikle en brugervenlig og funktionsdygtig hjemmeside, hvor kunder kunne skabe deres egne cupcakes ud fra et mangfoldigt udvalg af kombinerbare toppe og bunde, samt blive guidet igennem en brugervenlig betaling- og bestillingsprocess.

Projektets planlægningsfase involverede et mockup af hjemmesiden samt forskellige diagrammer og modeller, som fungerede som styrepinde for kodens opbygning i både front- og backend. Disse materialer dannede grundlaget for den efterfølgende implementering.

Projektet har til formål at demonstrere et professionelt arbejdsflow og give et realistisk billede af, hvordan en fuldstændt hjemmeside udvikles fra bunden.

Denne rapport er udarbejdet til en fagfælle - en datamatikerstuderende på 2. semester - med henblik på at give indsigt i hele udviklingsforløbet fra analyse og design til implementering, evaluering og refleksion over de erfaringer, der blev gjort undervejs.

BAGGRUND

Olsker Cupcakes er en Bornholm baseret bager med speciale i at lave cupcakes. De vil gerne have lavet en hjemmeside til at bestille deres varer, så kunderne kan få en god oplevelse hos dem. Et par københavnere har været forbi og indsamlet krav til hjemmesiden, som nu er til udviklernes opgave at lave en funktionel side med et lækkert design.

Olsker Cupcakes, kom selv med et mockup som kunne bruges som inspiration til deres hjemmeside, ses på **Bilag 1** Olsker cupcakes egen mockup af hjemmeside

KRAV

Olsker Cupcakes har givet 9 user stories, funktionelle krav, som lyder:

- US-1: Som kunde kan man bestille og betale cupcakes med en valgfri bund og top, sådan at man senere kan køre forbi butikken i Olsker og hente sin ordre.
- US-2 Som kunde kan man oprette en konto/profil for at kunne betale og gemme en ordre.
- US-3: Som administrator kan man indsætte beløb på en kundes konto direkte i Postgres, så en kunde kan betale for sine ordrer.
- US-4: Som kunde kan man se mine valgte ordrelinjer i en indkøbskurv, så man kan se den samlede pris.
- US-5: Som kunde eller administrator kan man logge på systemet med email og kodeord. Når man er logget på, skal man kunne se sin email på hver side (evt. i topmenuen, som vist på mockup'en).
- US-6: Som administrator kan man se alle ordrer i systemet, så man kan se hvad der er blevet bestilt.
- US-7: Som administrator kan man se alle kunder i systemet og deres ordrer, sådan at man kan følge op på ordrer og holde styr på sine kunder.
- US-8: Som kunde kan man fjerne en ordrelinje fra sin indkøbskurv, så man kan justere sin ordre.
- US-9: Som administrator kan man fjerne en ordre, så systemet ikke kommer til at indeholde ugyldige ordrer. F.eks. hvis kunden aldrig har betalt.

Database

Som det kan ses på diagrammet (**Bilag 2. ERD**), er der 6 tabeller, hvoraf der er; en user-tabel til at holde loginoplysningen for hver bruger, samt om det er en admin eller kunde, med en én-til-mange relation til en order-tabel. Order-tabellen indeholder hver ordres tidspunkt samt email på brugeren, der har bestilt, og har en én-til-mange relation til order_holder tabellen. Denne tabel fungerer som joint tabel til at holde på cupcakes samt antal, der skal tilføjes til en ordre. Derfor har den et order ID og et cupcake ID. Cupcake-tabellen er også en joint tabel der er sammenkoblet mellem topping og bottom. Dens opsætning gør det let lave en oversigt over alle kombinationer af toppings og bottoms og hvilke priser de har sammenlagt. Den får nemlig data fra topping og bottom tabellerne, som hver holder et navn og en pris som fx "Vanilla", "5" (kr).

Teknologier

Version Control → Git

Programmeringssprog → Java

Java Development Kit → v. 17

Integrated Development Environment → IntelliJ IDEA 2025.2.3 (Ultimate Edition)

Build Tool → Apache Maven 3.10.1

Database → PostgreSQL 42.7.2

Web Framework → Javalin 6.1.3

Template Engine → Thymeleaf 3.1.2

Frontend → HTML 5, CSS 3, JavaScript (ES6)

Særlige forhold

1. Hvad gemmes i sessionen?

Når en kunde logger ind, gemmer vi vedkommende som et objekt og lagrer toppings og bottoms i sessionen. Når kunden er logget ind, kan han/hun tilføje et vilkårligt antal cupcakes til sin kurv. Når personen går til checkout, gemmes kurven, indtil brugeren har betalt, hvorefter den bliver tømt.

Når en admin logger ind, gemmer vi vedkommende som et objekt og henter alle kunder fra databasen samt oplysninger om, hvilke køb der er blevet foretaget.

2. Hvordan blev exceptions håndteret?

Exceptions håndteres via *DatabaseException*, hvor brugeren får besked om, at han/hun ikke var i stand til at forbinde til databasen og hvorfor. Hvis brugeren ikke kan huske sin e-mail eller adgangskode, eller forsøger at registrere sig med en e-mail, der allerede er knyttet til en eksisterende bruger, håndteres dette ligeledes med passende fejlbeskeder.

3. Validering af brugerinput:

Når en bruger prøver at registrere sig selv i systemet, så tjekkes det om deres email indeholder et @, og om username er længere end 3 og kortere end 13 og om password er længere end 3 og kortere end 41

4. Hashing af password:

Når en bruger har besluttet sig om et password, og e-mailen ikke allerede har en bruger registreret, så hashes deres password via BCrypt, hvor der generes 12 runder af salt.

5. Brugertyper:

Systemet har mulighed for to brugertyper, en *customer* og *admin*, hvor ændring af brugertype skal manuelt blive ændret i postgres databasen.

Status på implementering

1. CRUD: Det er ikke muligt for en admin at fjerne brugere eller tidligere køb.
2. Fejl i systemet: En bruger kan gå til betaling, uden at have noget i sin basket.
3. I ordrer, kan man kun se cupcake id, da dato og tidspunkt ikke virker. I betaling, kan man under dato vælge tidligere tidspunkter, end selve dagen du er på.

Proces

Arbejdsprocessen var opdelt i tre forskellige samarbejdsformer.

1. I **“Code With Me”** blev der arbejdet i ét dokument, hvori man kunne kode og diskutere løsninger i realtid.
2. Derudover blev der brugt **branches**, så arbejdet kunne ske på forskellige funktioner uden at forstyrre hovedprojektet.
3. I den sidste arbejdsmetode blev der gjort brug af en **storskærm**, hvori 1 person koder, hvori resten observerer og kommer med input.



Planen med disse tre arbejdsformer var at udnytte fordelene ved både online og offline samarbejde.

Det som fungerede bedst, var faste mødetider hver dag, hvor man kunne sidde fysisk og have en dialog med hinanden om koden. Vi havde en dagsplan, som blev kreeret på 3-dages basis. Som vi fulgte til vejs ende, så vi hver dag vidste hvor langt i processen vi var nået.


Selve arbejdsfordelingen gik godt, vi fik nået det vi skulle hver dag, dog brugte vi ikke en scrum master, hvilket skabte nogle udfordringer ift. at man ikke nødvendigvis vidste hvor langt hver person var nået. Dette var en udfordring især på de dage, hvor man ikke var i stand til at møde op fysisk.

BILAG

Bilag 1 Olsker cupcakes egen mockup af hjemmeside



OrdreKunder

jobe@cphbusiness.dk

Velkommen ombord

Øens bedste cupcakes. Vælg og bestil her:

Vælg bund

Vælg topping

Vælg antal

Læg i kurv

8

Bilag 2. ERD

