

Cupcake

Malde Frank Midtgaard, cph-mm737@cphbusiness.dk, github.com/Mald004
Julius Baldrian Lansner, cph-jl483@cphbusiness.dk, github.com/JuliusLansner
Mikkel Kern Hasselmark, cph-mh1088@cphbusiness.dk, github.com/cph-mh1088

Copenhagen Business Academy, 30 Marts 2023
DAT1-E22, Klasse A, Gruppe 5

Indhold

Indledning.....	3
Baggrund.....	3
Teknologivalg.....	3
Krav.....	4
Aktivitetsdiagram.....	5
EER diagram.....	6
Navigationsdiagram.....	6
Særlige forhold.....	7
Status på implementation.....	7
Process.....	8
Video-demo.....	9

Indledning

Denne rapport dokumenterer 2. semester projektet “Cupcake” på datamatikerstudiet. Målgruppen for denne rapport er datamatiker studerende på 2. semester. Projektet omhandler en fiktiv kundes ønske til en hjemmeside, hvor hans kunder kan lave bestillinger på cupcakes. Med udgangspunkt i en startkode, skal der laves både frontend og backend, indtil man har en funktionel hjemmeside.

Baggrund

Virksomheden Olsker Cupcakes er et iværksætterprojekt fra Bornholm, som producerer økologiske cupcakes. De kunne godt tænke sig en hjemmeside, hvor deres kunder kan oprette bestillinger. Virksomheden har fremstillet en mockup af en tænkt forside. Det første kundemøde resulterede i ni såkaldte user-stories, der kort beskriver hjemmesidens ønskede funktioner. Nedenstående er kundens krav til systemet.

Som kunde skal man have følgende muligheder på hjemmesiden:

- Man kan oprette sig som bruger og gennemføre en ordre.
- Man kan placere en ordre på cupcakes med valgfri top og bund til afhentning.
- Man kan tilføje cupcakes indkøbskurven, og løbende se sin totale pris.
- Man kan fjerne et produkt fra sin indkøbskurv, og dermed justere sin ordre.
- Når man er logget ind skal man kunne se eget brugernavn på hver side.

Som administrator skal man kunne følgende på hjemmesiden:

- Man skal kunne indsætte et beløb ind på en kundes konto.
- Når man er logget ind, skal man kunne se eget brugernavn på hver side.
- Man skal kunne se alle ordre der er blevet placeret.
- Man skal kunne se alle kunder i systemet, samt deres ordre.
- Man skal kunne fjerne en ordre, så systemet ikke indeholder ugyldige ordre.

Teknologivalg

I projektet er der gjort brug af følgende teknologier:

- Java Amazon Corretto 11
- MySQL version 8.0.31
- Bootstrap 4.0
- IntelliJ 2021.2.4
- Apache Tomcat 9.0.73
- HTML5 + CSS
- Jakarta EE (Java Enterprise)

Krav

Nedenstående er de ni userstories fra det første kundemøde:

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 MySQL, 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.

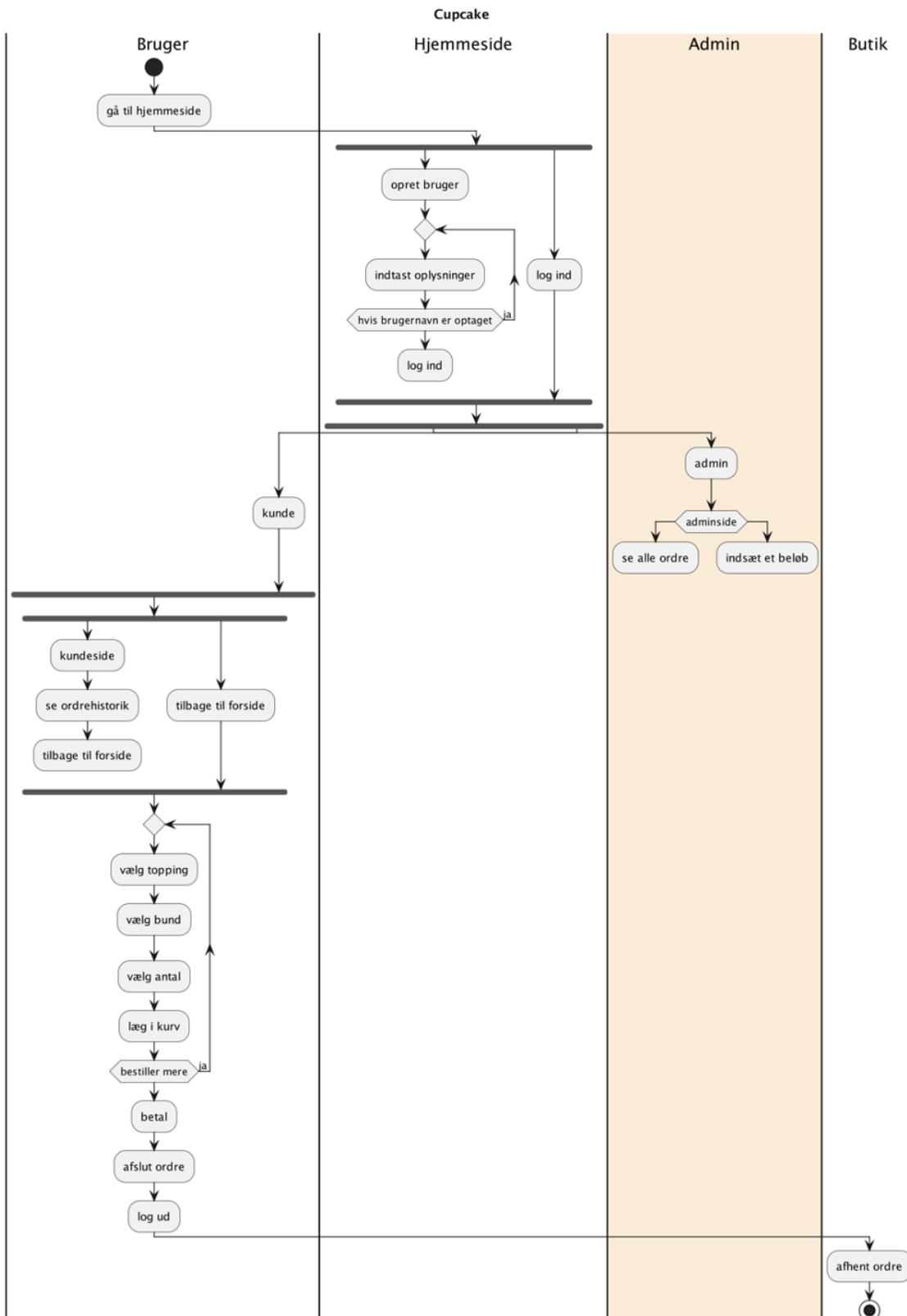
US-7: Som administrator kan jeg se alle kunder i systemet og deres ordrer, sådan at jeg kan følge op på ordrer og holde styr på mine kunder.

US-8: Som kunde kan jeg fjerne en ordre 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.

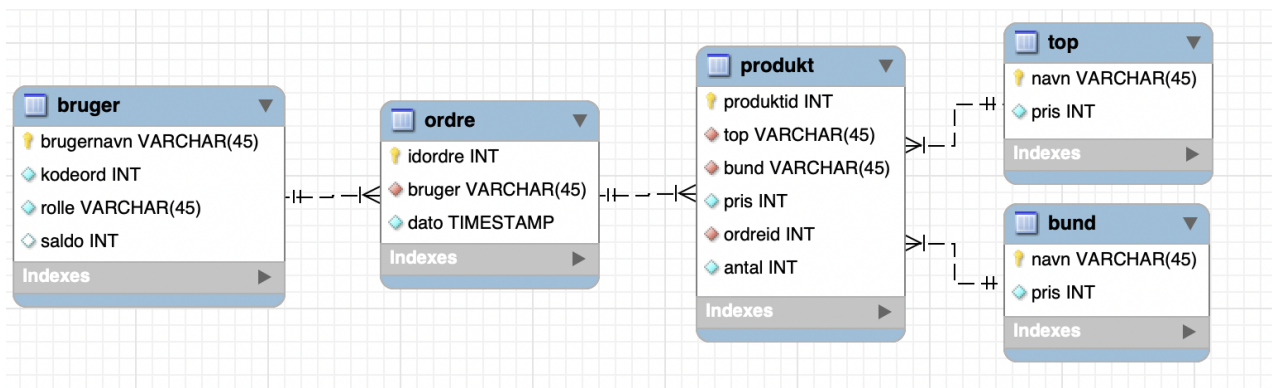
Aktivitetsdiagram

Nedenstående er et “AS-IS” aktivitetsdiagram med svømmebaner for vores cupcake hjemmeside.



EER diagram

Her ses et enhanced entity-relationship (EER) diagram over vores database.

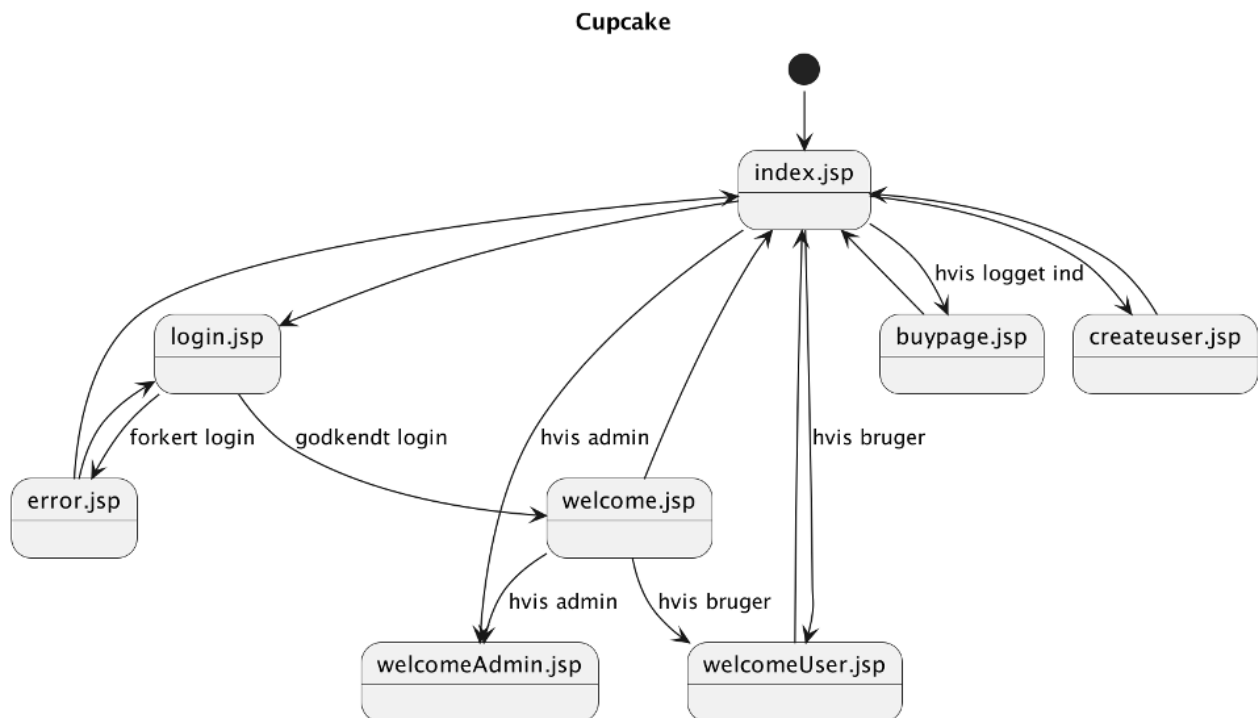


Databasen er i første form, da der ikke bruges row order til at udtrykke info, der er ikke mikset datatyper, alle tables har en primary-key og der er ikke nogen rows som har flere grupper af data på sig.

Og det er så i tredje form, da alle non-key attributter afhænger af en primary-key i hvert table og intet andet.

Navigation diagram

Nedenstående ses et “AS-IS” navigation diagram over vores hjemmeside.



Brugeren lander først på index.jsp, hvor man enten skal logge ind, eller oprette sig som bruger for at kunne komme videre. Når man er logget ind, kan man placere en ordre på index.jsp, og derefter kan man betale på buypage.jsp. Afhængigt af brugerens rolle kan man også tilgå en personlig side. En bruger med rollen “user”, kan tilgå welcomeUser.jsp, hvor man har mulighed for at se tidligere ordre. En bruger med rollen “admin”, kan tilgå welcomeAdmin.jsp, hvor man har mulighed for at se alle ordre, eller indsætte et beløb på hos en bruger.

Særlige forhold

I session gemmes følgende informationer:

Ordre indhold, kurv indhold, en boolean der giver true/false angående logged in status, en cupcake top liste, en bottom cupcake liste og den nuværende logged in users brugernavn, kodeord og rolle.

Vores exceptions handles med DatabaseException, hvor vi viser printStackTrace.

Vores brugerinput bliver valideret via en html form, hvis data bliver sat imod vores database. Hvis brugernavn + kodeord kombo eksisterer, kan brugeren logge ind.

Sikkerheden af et login er lavet sådan, at en bruger ikke kan få adgang til eller se brugernavn/password af andre brugere via hjemmesiden, men det er derimod stored i vores database som kun en “admin” ville have adgang til, ift. en tekstfil der ligger i hjemmesidens kode/opbygning.

Bruger typerne vi har er “user” og “admin”, f.eks i funktionen hvor man kan klikke på sit brugernavn, for at komme til en specifik side, hives der på rollen for at fortælle en servlet hvilken side der skal tilgås, en users side eller admin siden.

Status på implementation

Status pr. 30/3 - 2023 er at hjemmesiden er funktionel, og de første otte user-stories er implementeret. Hjemmesiden er dynamisk, og okay responsiv. For at sætte de korrekte størrelser kunne vi have brugt @media screen and (min/max) for at forbedre menuerne på de forskellige størrelse af skærme.

På nuværende tidspunkt kan alle nyoprettede brugere tildele sig deres egen rolle. Ved evt. videreudvikling, ville det være en af vores førsteprioriteter at få fikset.

Metoderne indsætBeløb() og updateSaldo(), som findes i userMapper virker begge i main programmet, men deres test fejler.

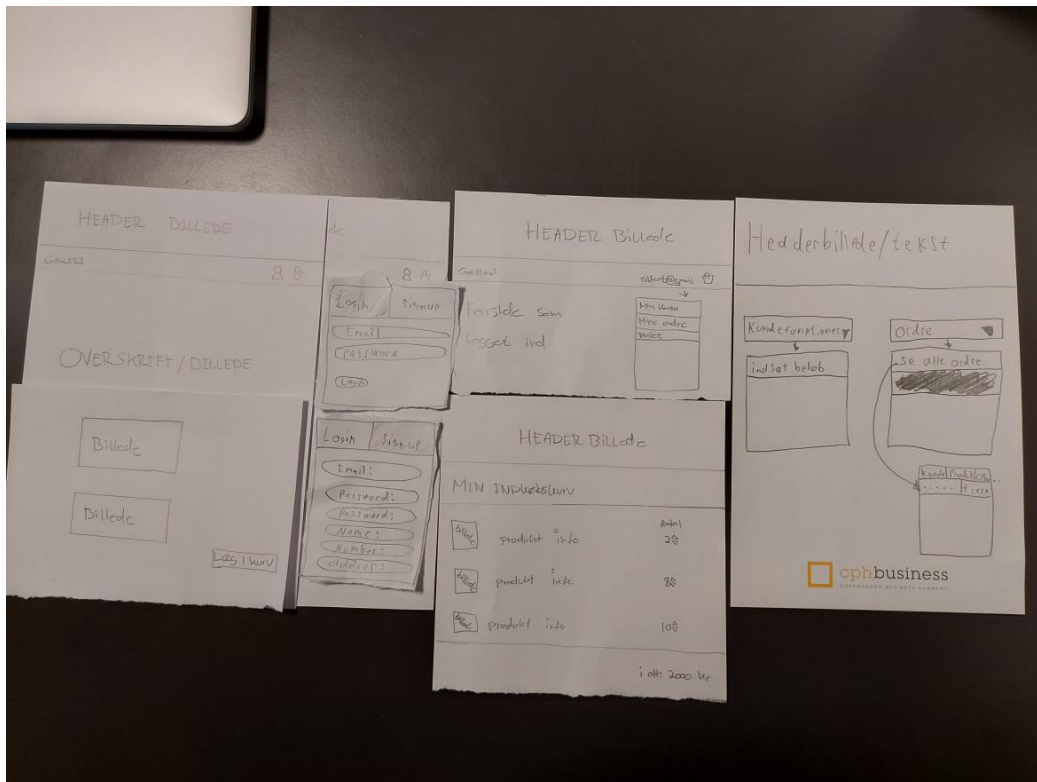
Process

1. Planen.

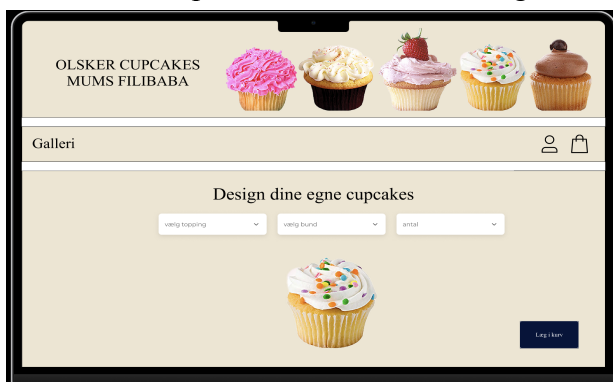
Vores plan var at starte med at lave noget visuelt, som vi kunne tage udgangspunkt i, og derefter uddelegere opgaver.

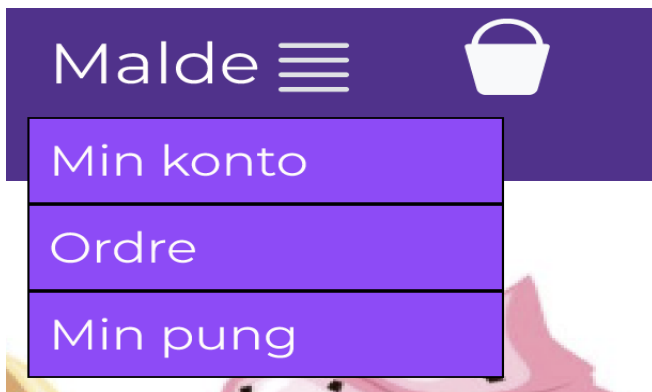
2. Hvordan det gik.

- Vi startede med at tegne en skitse af hvordan vi ville have det:



- Derefter lavede vi hver især en Figma af hvordan vi syntes det skulle se ud, også valgte vi dele af præsentationerne som vi godt kunne lide og samlede det:





- Vi lavede samlet en database.
- Der blev uddelegeret opgaver, nogle gik i gang med frontend og nogle gik i gang med backend. Vi begyndte at designe, og vi fik lavet mappere og facader med de metoder vi mente vi skulle bruge.

3. Perspektivering.

Alt i alt synes vi alle det gik meget godt, vi fik lavet en god indgangsvinkel til vores projekt med vores Figma præsentation, og vi var alle meget med på hvad der skulle ske og hvordan det skulle se ud.

4. Hvad vi har lært.

Vi har været meget glade for at have en god Figma præsentation at kunne kigge på, i det at vi alle fik en samlet forståelse af hvordan tingene skulle hænge sammen, det har hjulpet os en del hen af vejen, og lært os om hvor vigtigt det kan være at strukturerer ens projekt. Vi har også lært rigtig meget om brugen af servlets, web udvikling med css og html(jsp) og øget forståelsen for brug af en MySQL database.

Video-demo

<https://youtu.be/yjyb5JrQvjM>