

Cupcake-projekt

Navn: Anders og Lukas

cph-email: cph-al350@cphbusiness.dk og cph-ln288@cphbusiness.dk

github-navn: AndersLind99 og Cjapz

Klasse: Bornholm studerende

Tidspunkt: Fredag d. 16 April 2021 - Onsdag d. 28 April 2021

Indholdsfortegnelse	
Indledning:	3
Baggrund:	3
teknologivalg:	3
Krav:	3
Diagrammer	4
aktivitetsdiagram:	4
Domæne model:	4
ER diagram:	6
navigationsdiagram:	7
særlige forhold:	7
Status på implementation:	7
proces:	8

Indledning:

Denne opgave omhandler en case vi har fået stillet af Olsker Cupcakes, som er et Cupcake firma på Bornholm. Olsker cupcakes ønsker en nem hjemmeside til at kunne modtage bestillinger fra kunder.

Baggrund:

Cupcake firmaet Olsker Cupcakes, ønskede en nem hjemmeside til deres kunder så de kunne modtage bestillinger. Det var vigtigt for Olsker Cupcakes at kunden skulle kunne vælge både en top, en bund og hvor mange cupcakes de vil have. Websitet skulle være nemt og bruge både for kunder og medarbejdere.

teknologivalg:

intellij ultimate 2021.1
intellij ultimate 2021.0
java 1.8
tomcat 9.0.45
adobe creative cloud
mySQL workbench 8.0 CE
jdbc
Maven 4.0
bootstrap 4.3.1

Krav:

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.

Som kunde kan jeg oprette en konto/profil for at kunne betale og gemme en ordre.

Som administrator kan jeg indsætte beløb på en kundes konto direkte i MySQL, så en kunde kan betale for sine ordrer.

Som kunde kan jeg se mine valgte ordrelinjer i en indkøbskurv, så jeg kan se den samlede pris.

Som kunde eller administrator kan jeg logge på systemet med email og kodeord. Når jeg er logget på, skal jeg kunne min email på hver side (evt. i topmenuen, som vist på mockup'en).

Som administrator kan jeg se alle ordrer i systemet, så jeg kan se hvad der er blevet bestilt.

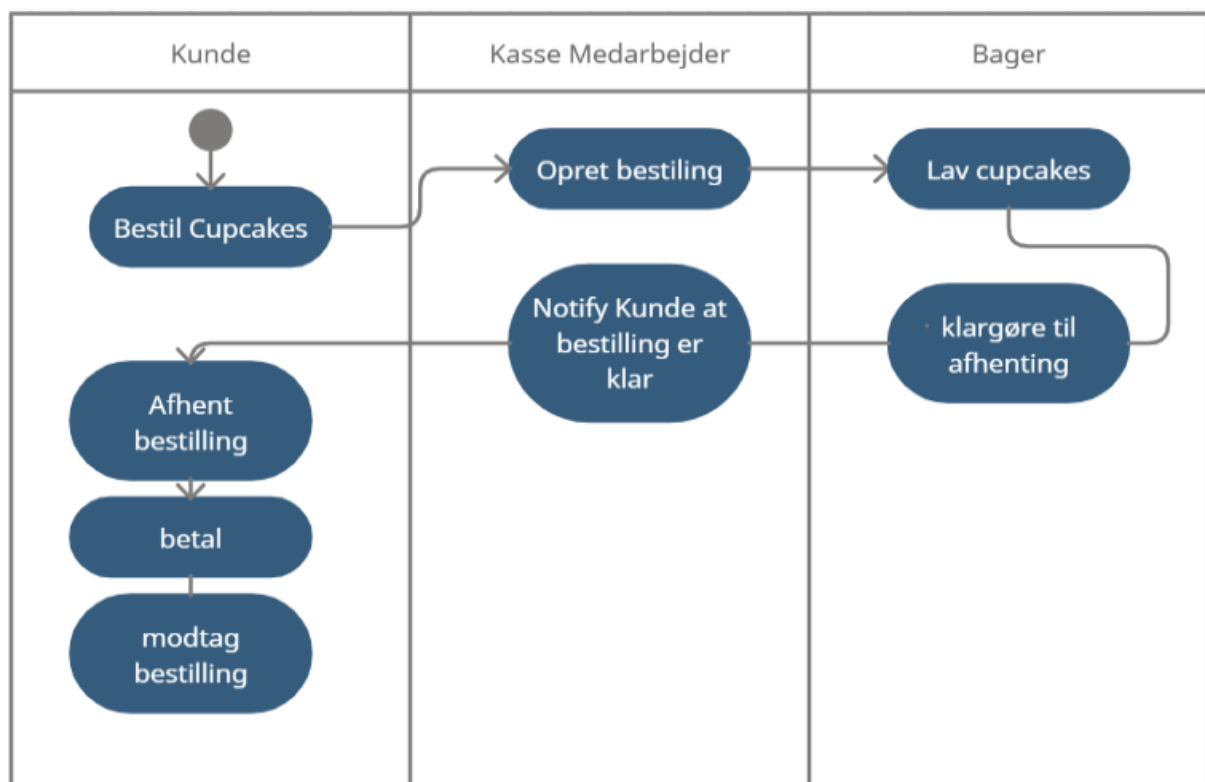
Som administrator kan jeg se alle kunder i systemet og deres ordrer, sådan at jeg kan følge op på ordrer og holde styr på mine kunder.

Som kunde kan jeg fjerne en ordre fra min indkøbskurv, så jeg kan justere min ordre.

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

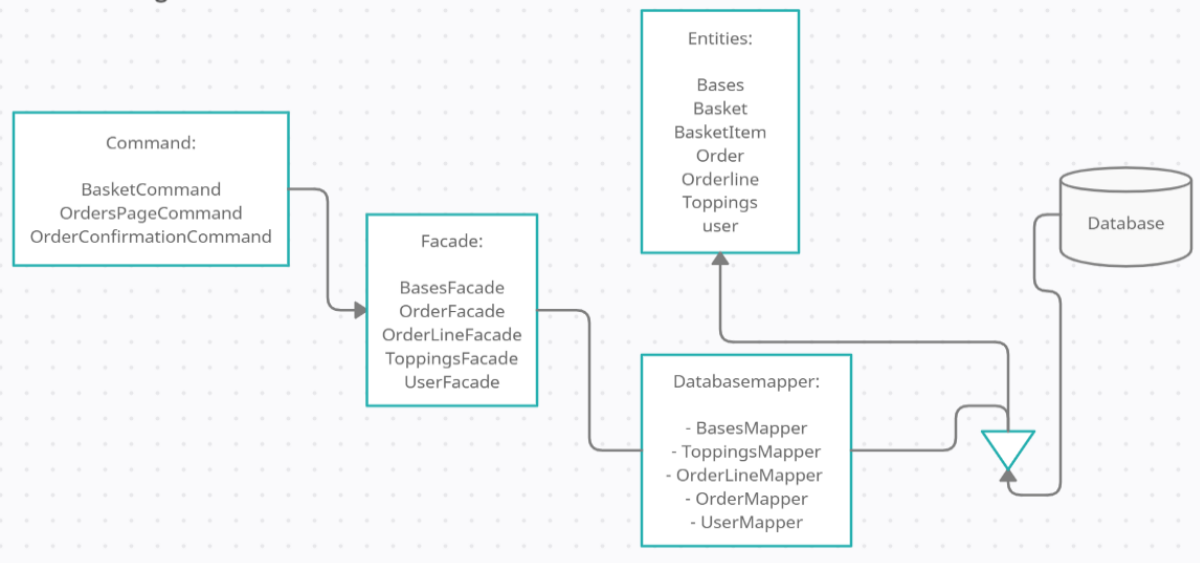
Diagrammer

aktivitetsdiagram:



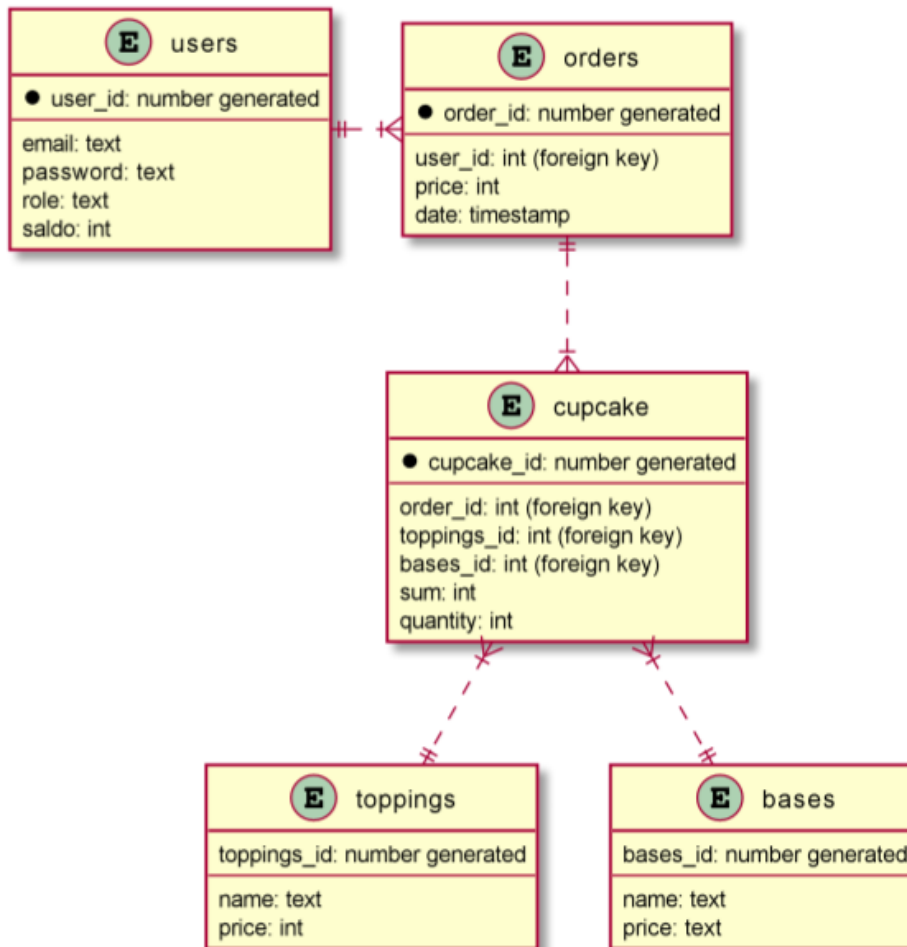
Domæne model:

Domæne Diagram



vores *basket* og *basketItem* er den vi har måtte overveje mest. hvordan vi skulle lave vores basket, og fylde den op på en enkel og simpel måde. der ville gøre den nemmere at arbejde med i resten af koden.

ER diagram:

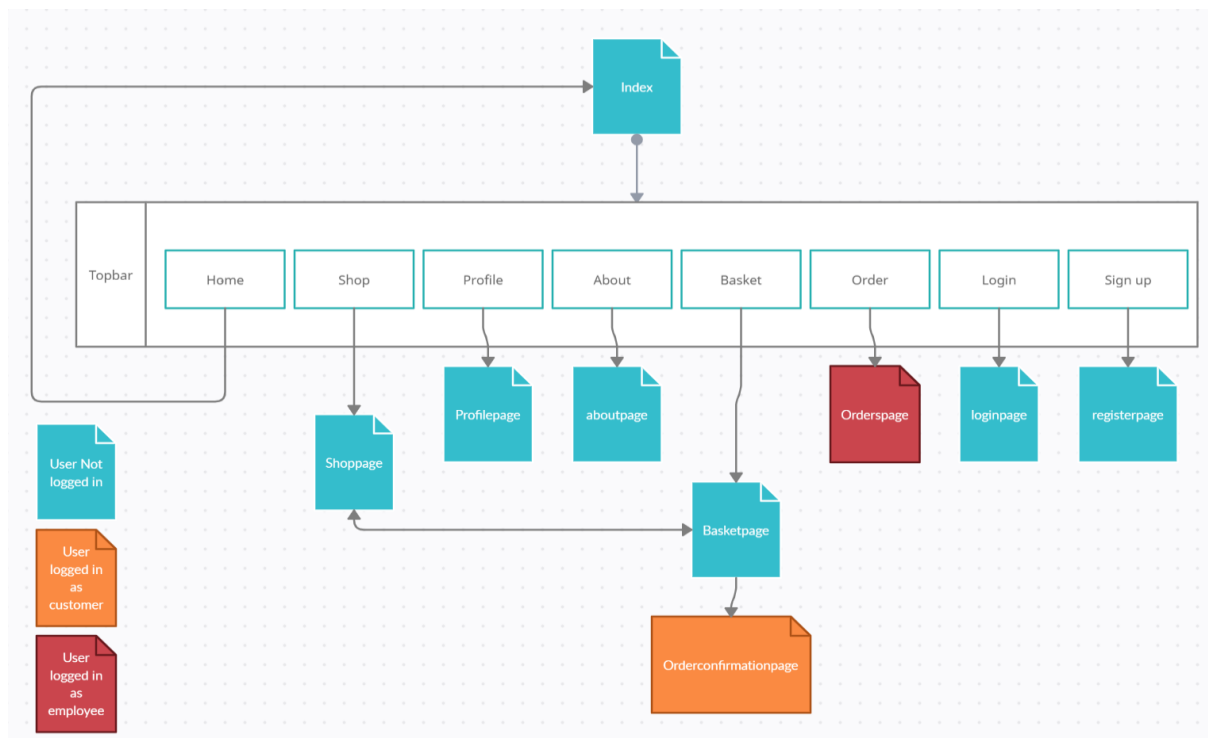


Vores tanke bag vores ER diagram var meget simpelt, en ordre skulle kunne indeholde mere end 1 type cupcake, da det ikke ville være særlig User Friendly hvis de skulle oprette flere bestillinger når de ville bestille.

cupcaken skulle kunne bestå af alle de forskellige kombinationer af toppings og baser, og have mulighed for at blive nemt udvidet i fremtiden så de blev delt op i deres egne tabeller. hvor vi så bruger deres ID til bygge vores cupcake.

vi skulle vide hvem der havde bestilt hvad, derfor puttede vi et user ID på vores ordre, på den måde har vi nu en database som indeholder *USERS* som kan oprette *ORDERS* hvor dens *ORDER_ID* vil stå på vores *CUPCAKE* så vi ved hvilken order den hænger sammen med og vi har vores *CUPCAKE* som består af et *BASES_ID* & *TOPPINGS_ID* så vi ved hvilken slags cupcake der er blevet lavet. vi kunne muligvis have fjernet *SUM* fra vores *CUPCAKE* men vi tænkte det var bedre at kunne se summen på vores *CUPCAKE* direkte i vores Tabel.

navigationsdiagram:



vi har benyttet os af en fælles navigation bar i toppen for at gøre det nemmere og mere overskueligt for brugeren at navigere sig rundt.

vi har sat de sider som kun kan tilgås af brugeren når de er logget ind som "customer" i orange og de sider der kun kan tilgås af brugeren når de er logget ind som "employee" i rød. resten i blå kan tilgås uanset om man er logget ind eller ej.

særlige forhold:

Vi gemmer basket informationen i session.

vi har lavet en brugertype i vores Database, som går ind og har fuld kontrol til at lave ændringer i databasen. dette ville vi gerne splitte op til 2 database bruger, en som har adgang til at oprette ordrer og en anden admin bruger som kan gå ind justere saldo på bruger og lave andre ændringer diverse steder hvis nødvendigt.

Status på implementation:

- Styling mangler på hele websiden
- Ordrelisten mangler at kunne vise hvad en ordre består af, så den ikke kun viser et user_id, total pris og dato for oprettelsen.
- flere bruger i databasen, så admin brugeren ikke bliver brugt til når det en customer som er logget ind.
- mulighed for at fjerne ting fra basket.
- mulighed for at slette ordrer.

proces:

Vi blev enige som en gruppe om en simpel aftale der hed at vi mødtes hver dag klokken 10:00. Derfra vil vi kode til vi ik kunne mere, hvilket typisk endte op i vi kode 6 timer om dagen.

Vi lykkedes med at få lavet de første 6 US. Dertil lavede vi også alle de ikke-funktionelle krav samt Deliveries. Der er nogle fejl og mangler i de manglende 3 US.

Det vi syntes gik godt var at, vi følte vi nåede rigtig langt selvom vi kun var to i gruppen. Vi har fundet ud af vi er et godt team og kan hjælpe hinanden til at løse mulige fejl og problemer med koden. Vi har hver vores styrker og svagheder, hvilket gjorde et godt samarbejde, da vi kunne bruge vores styrker der hvor den anden havde svagheder.

Det der kunne være gået bedre var at, vi første dag brugte måske lidt for meget tid på databasen der ik vil fungere og vores adobe mockup. Efter vores første fejl med brug af arbejdskraft, følte vi ik rigtigt der var noget vi kunne have gjort bedre, med den tid vi havde.

Til næste gang vil vi først og fremmest være en større gruppe. Når vi skal snakke fremgang i arbejdet, vil vi nok sætte vores fokus et andet sted fra start. Derfra kan vi samle op der hvor vi skulle møde eventuelle fejl, frem for at bruge for lang tid på at løse fejl som måske ik eksistere.