28-04-2021

Cupcake

Deltagere (navn, email, Github navn):

Klasse A

- Abdi Abdulle, cph-aa394@cphbusiness.dk, abdimo101.
- Mikkel Gottschalck, cph-mg320@cphbusiness.dk, mikkeg05.
- Nicolai Martini, cph-nm172@cphbusiness.dk, nmartini17.
- Thomas Amorsen, cph-ta181@cphbusiness.dk, cph-ta181.

Indledning

Cupcake projektet er et projekt hvori vi er blevet stillet med en opgave om at designe en hjemmeside og en tilhørende database der skal bruges til at bestille cupcakes.
På hjemmesiden skal der kunne oprettes konti til kunder, sådan at de kan bestille cupcakes med valgfri bund og top, så kunden senere hen kan tage til butikken og hente deres cupcakes, uden at skulle stå i kø og vente.
Databasen skal indeholde de bunde og toppings der kan vælges mellem samt indeholde konti for kunder og ansatte.
Kunder skal kunne ændre i deres indkøbskurv, sådan at hvis der noget som de ikke gider have alligevel, så kan de slette det fra deres kurv.
De ansatte skal kunne indsætte penge på kunders konti, så de kan betale for deres cupcakes, herudover skal de ansatte kunne se en liste over hvilke kunder har bestilt hvilke cupcakes og dermed have styr på henholdsvis ordrer og kunder.
Ansatte skal også kunne slette ordrer, som ikke er blevet betalt.
Dette har vi opnået ved brug af Java, SQL, JDBC til at forbinde de to og Intellij som IDE.

Baggrund

Olsker Cupcakes er et bornholmsk iværksætteri som går op i økologi.
Som kunde kan man bestille cupcakes med en valgfri bund og top samt betale for den, sådan at kunden kan tage forbi Olsker Cupcakes og hente deres ordre.
Som kunde kan jeg oprette en bruger på hjemmesiden og betale samt gemme en ordre til senere. Kunden skal også kunne se hvad der er lagt i indkøbskurven, så kunden kan se den samlede pris, samt skal kunden også kunne fjerne og redigere sin egen ordre.
Som administrator skal man kunne sætte penge ind på kundernes konto så de kan betale for en ordre, samt kunne se alle ordrer i systemet. Udover at kunne se alle ordrer i system så skal administratoren have mulighed for at fjerne ordrer, så der ikke er ugyldige ordrer i systemet og det skal være muligt for administratoren at se hvilke kunder bestiller hvad, så der kan følges op på ordrerne og holde styr på kunderne.

Teknologivalg

IntelliJ IDEA 2021.1 (Ultimate Edition)

Build #IU-211.6693.111, built on April 6, 2021

Runtime version: 11.0.10+9-b1341.35 amd64

VM: Dynamic Code Evolution 64-Bit Server VM by JetBrains s.r.o.

Kotlin: 211-1.4.32-release-IJ6693.72

MySQL

MySQL Workbench 8.0

Version 8.0.22 107600 CE (64 bits)

JDBC

8.0.19

JAVA

1.8

Krav

Firmaets håb med dette system

Firmaet vil gerne forøge brugervenligheden og gøre deres produkter let tilgængeligt for kunder som gerne vil bestille i forvejen. Derudover så bliver det også lettere når firmaet begynder med at levere deres produkter ud til kunder.

Userstories

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 min email på hver side (evt. i topmenuen, som vist på mock up'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.

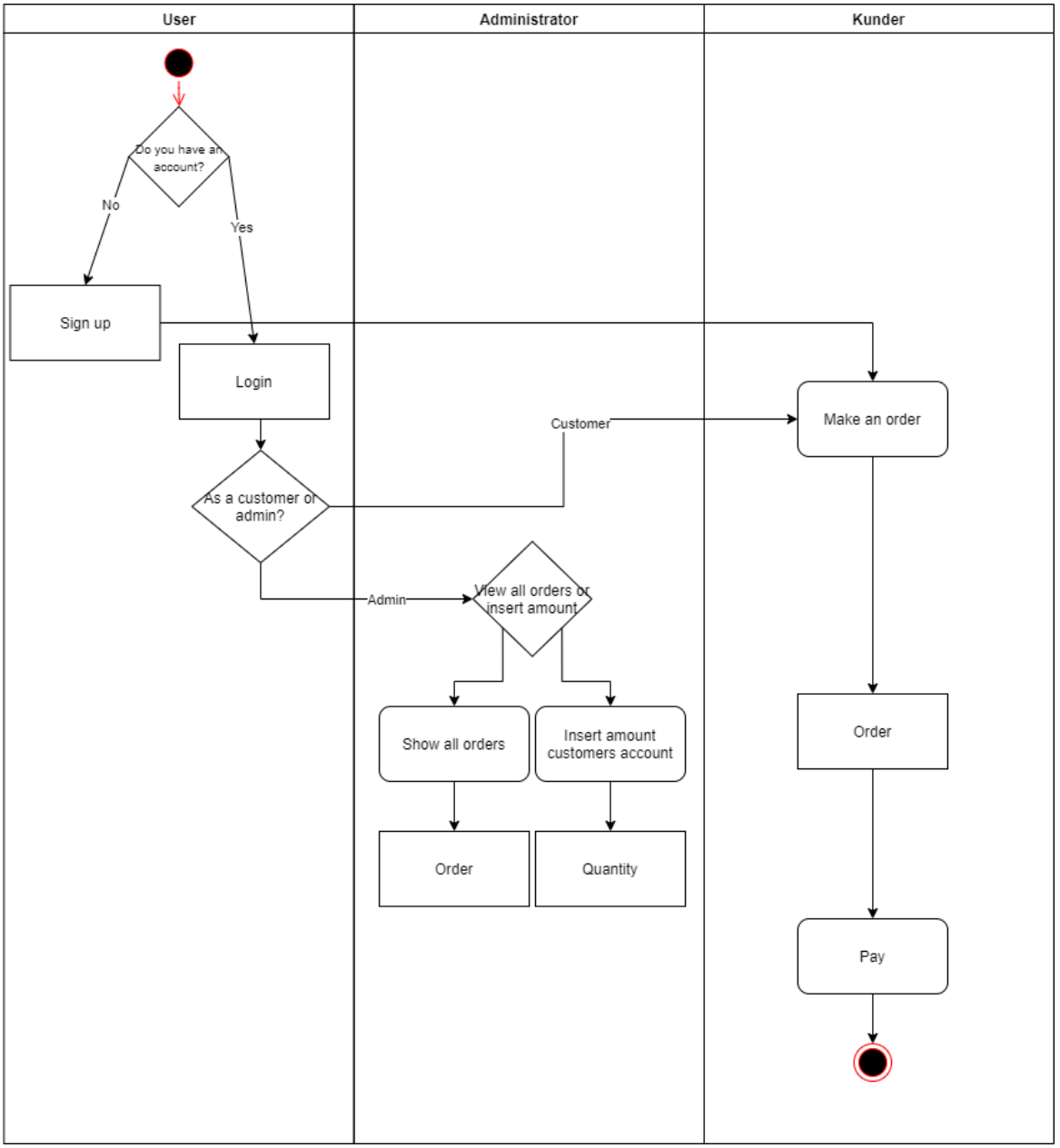
US-8: Som kunde kan jeg fjerne en ordre fra min indkøbskurv, så jeg kan justere min ordre.

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

I vores TO-BE aktivitetsdiagram starter vi som *user*, lige når man kommer ind på hjemmesiden, herefter bliver man nødt til at lave en bruger eller logge ind for at bestille. Når man har lavet en bruger og logget ind som *kunde*, så kan man gå i gang med at bestille cupcakes.

Hvis du logger ind som *administrator* kan du se alle ordrer og indsætte penge ind på *kundens* konto, så de kan betale for deres cupcakes.

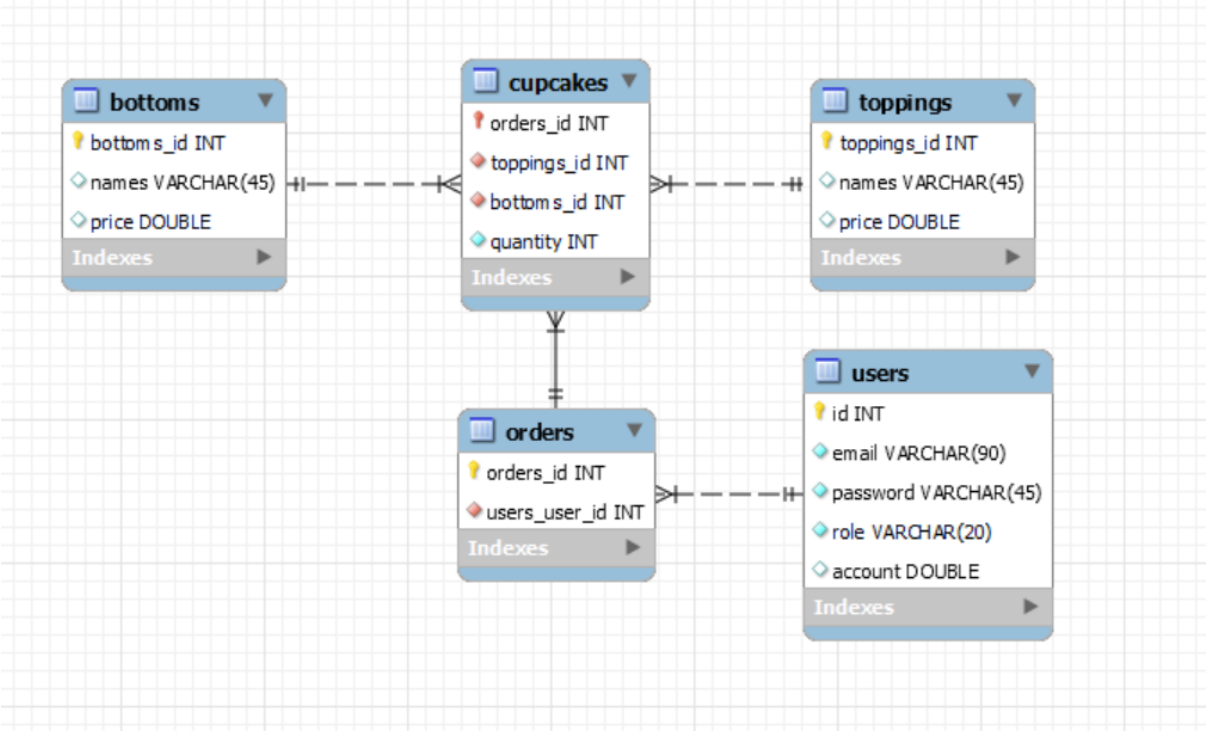


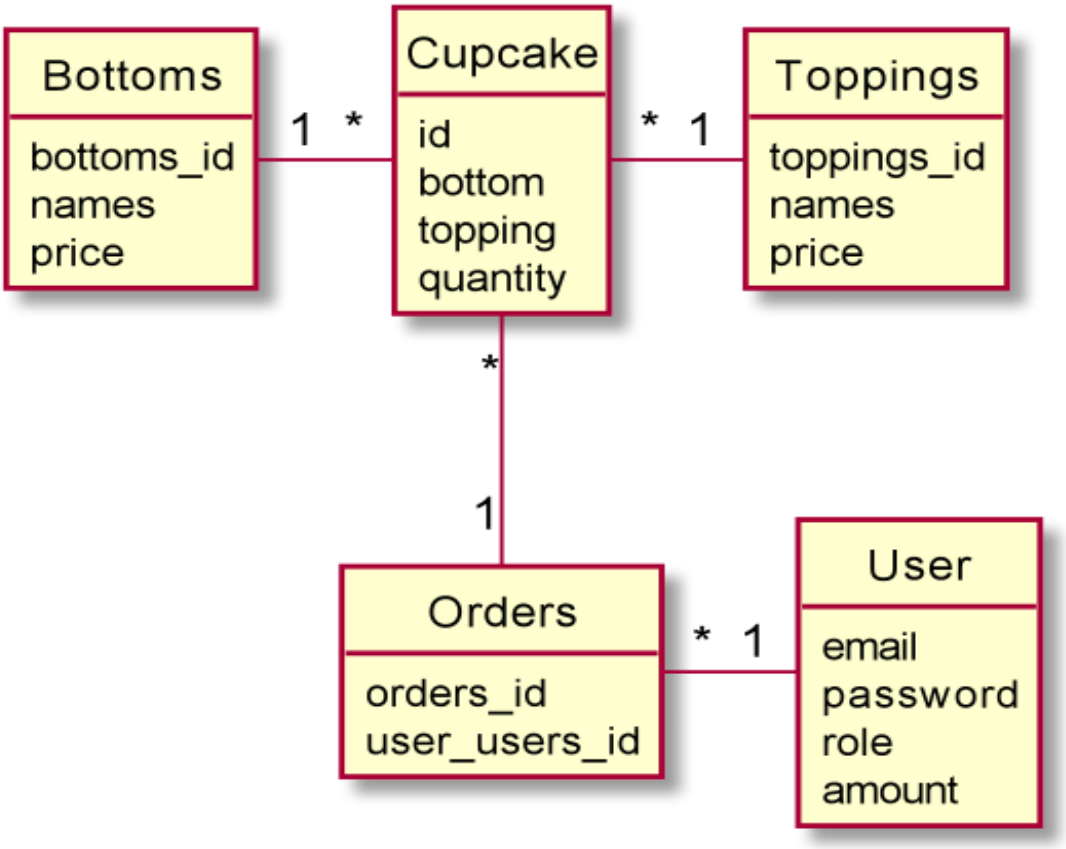
Domæne model og EER diagram

Domæne model og EER diagram:

Vores database har 5 tabeller, den første tabel er bottoms. Bottoms indeholder bottoms_id (primærnøgle), names og price. Den anden tabel er cupcakes og den indeholder orders_id (primærnøgle), toppings_id, bottoms_id og quantity. bottoms_id under cupcakes har en M-1 relation med bottoms_id i tabellen bottoms. Den tredje tabel er toppings og den indeholder toppings_id (primærnøgle), names og price. Primærnøglen under toppings har en 1-M relation med toppings id i tabellen cupcakes. Den fjerde tabel er orders og den indeholder orders_id (primærnøgle) og user_user_id. Primærnøglen i tabellen orders har en 1-M relation med tabellen cupcakes primærnøgle orders_id. Den femte og sidste tabel er users. Users indeholder id(primærnøgle), email, password, role og account. Primærnøglen har en 1-M relation med users_user_id som er under tabellen orders.

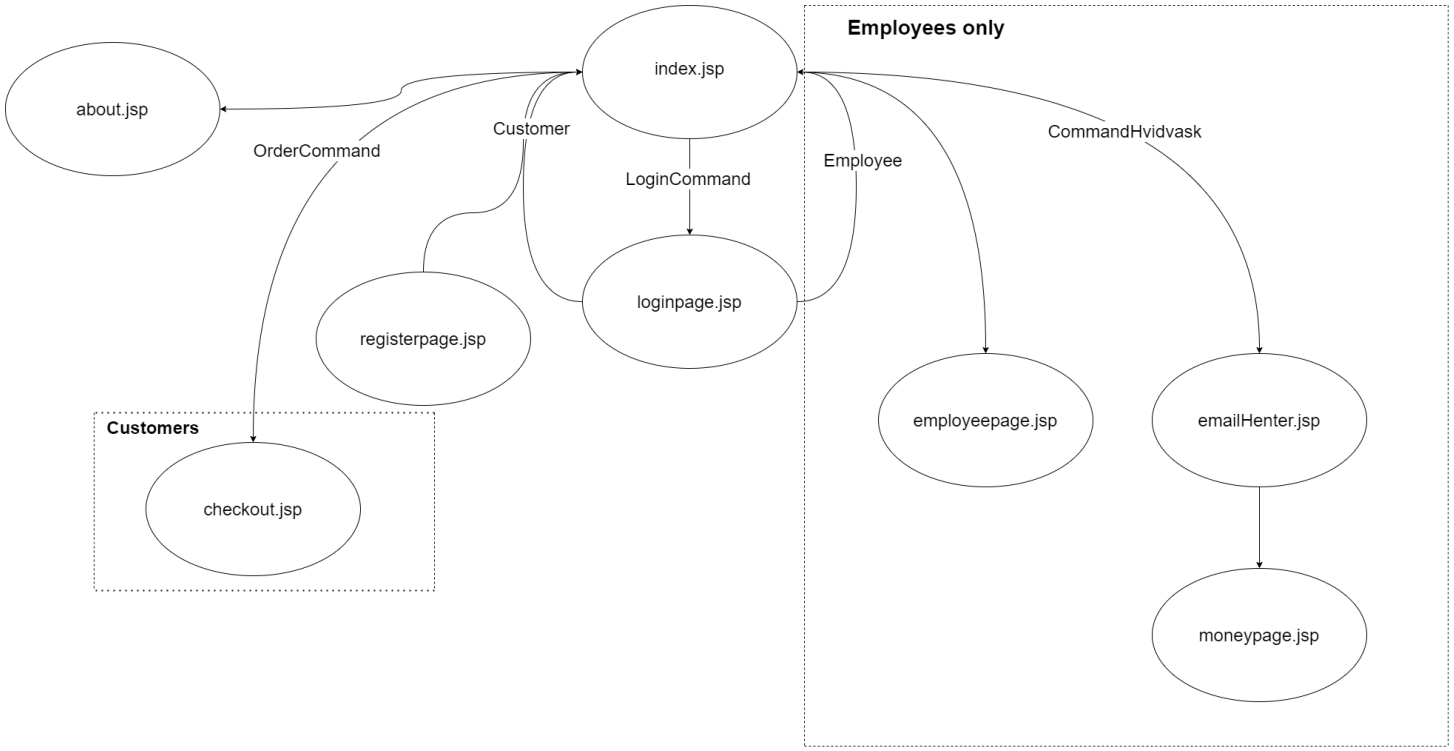
Den er på 3 normalform da attributterne udenfor primærnøglen ikke er afhængige med hinanden. Den eneste primærnøgle som ikke er autogenereret er orders_id som er under tabellen cupcakes. Grunden til det er at den henter data fra orders tabellen primærnøgle.




Bottoms
bottoms_id
names
price
1
*
Cupcake
id
bottom
topping
quantity
*
1
Toppings
toppings_id
names
price
*
1
Orders
orders_id
user_users_id
*
1
User
email
password
role
amount

Navigationsdiagram

I vores navigationsdiagram har vi benyttet os af en fælles navigations bar, fordi det gør det nemmere for kunderne at navigere rundt på siden. Ved brug af navigationsbaren gør vi det nemt for kunderne at hoppe rundt på siden ud at skulle klikke for mange gange, som hvis der har været en hamburger menu.



Særlige forhold

Informationer der gemmes i session:

I sessionen gemmes den bruger der på nuværende tidspunkt er logget ind på siden.

Exceptions håndteres ved:

Når en fejl opstår bliver der kastet en UserException hvis det sker på brugersiden og en SQLException når der sker en database-relateret fejl.
Her udskriver hjemmesiden en fejlmeddelelse der fortæller hvad der er gået galt.

Validering af brugerinput:

Det eneste sted brugerinput valideres er i forbindelse med login. Her tjekker siden hvorvidt det input der modtages stemmer overens med en bruger der allerede findes i databasen.

Sikkerhed i forbindelse med login:

Når man logger ind tjekker siden hvorvidt det input der modtages stemmer overens med en eksisterende bruger i databasen.

Status på implementation

Vores gruppe fik lavet de 6 første user stories. Men idet 6 userstories, så mangler gruppen at lave en funktion der kan vælge forskellige antal af cupcakes man gerne vil bestille i US-1. Udover det mangler vi i US-3 at lave en funktion der kan trække penge fra kunders konti, når de bestiller cupcakes. Derudover mangler vores gruppe også at lave en funktion der kan vise kundens valgte ordrelinje, da den kun kan vise den seneste ordre kunden har bestilt. Vi fik heller ikke brugt alle CRUD metoderne til alle tabeller. For eksempel brugte vi CREATE og READ metoderne i alle tabeller. UPDATE metoden blev brugt i *users* tabellen. DELETE metoden fik vi ikke brugt. Vi fik heller ikke brugt bootstrap til vores dropdowns fordi det virkede ikke.

Proces

Hvad var jeres planer for teamets arbejdsform og projektforløbet?
Vores gruppe startede med at bruge Kanban metoden ved brug af Github Issues til at gøre arbejdsprocessen mere overskueligt. Vi har også brugt Discord som kommunikationsværktøj.

Hvordan kom det til at forløbe i praksis?
Vi arbejdede normalt fra kl 10 til 14, men det variere lidt fra dag til dag. Desuden har vi også arbejdet på projektet som en enhed dvs. Der var en der delte sin skærm, mens de andre hjalp over Discord og det gjorde vi skiftevis. Det største problem i projektet var databasen, da vi havde problemer med at indsætte data fra hjemmesiden til tabellen *cupcakes*. På grund af det, har vi brugt en masse tid på at løse problemet end at lave de andre funktioner som er blevet nævnt under **status på implementation**.

Hvad gik godt og hvad kunne have været bedre?
Vi synes vores gruppearbejde fungeret godt, da alle møder op til tiden og bidrager til at få løst de user stories der er blevet givet. Vi føler at det var svært at få hjælp fra vejledere hver gang vi havnede i en uforståelig situation.

Hvad har I lært af processen og hvad vil I evt. gøre anderledes næste gang?

Næste gang vil vi prioritere vores tid på at opfylde de user stories der er blevet stillet først og gemme styling til sidst. Vi har tænkt os at gøre bedre brug af kanban board, da det ville give os et bedre overblik over hvad der mangler at blive implementeret.