Andreas Grand: cph-ag198@cphbusiness.dk, Git: ElGrand
Christoffer Perch Nielsen: cph-cn332@cphbusiness.dk , Git: Christoffer-Nielsen-Cph
Oliver Sloth Jensen: cph-oj101@cphbusiness.dk , Git: Oliverden3
Jonathan Braad:cph-jb442@cphbusiness.dk, Git: jonathanbraad
Cupcake projekt 21/4-2022
https://github.com/Datamatiker-2semester/Cupcake_Project

Olsker Cupcake Projekt

Youtube link til demo: https://youtu.be/LTeycfq2Dz0

Indholdsfortegnelse

Indledning:

I dette *cupcake* projekt har vi skulle lave en betaling-bestilling hjemmeside ved hjælp af primært Java, mysql og Tomcat. Ideen med hjemmesiden er, at man skal kunne have en bruger, og at man kan bestille cupcakes med forskellige toppings og bottoms. Som kunde skal man også kunne se ens saldo på hjemmesiden. Som administrator skal man have adgang til ordrer og kundernes oplysninger.

Baggrund:

Nogle hipsters fra København ville gerne lave en hjemmeside deres cupcake business, hvor de har lavet en halvfærdig mockup af en tænkt forside.

I dette projekt skal kunden kunne gøre nogle specifikke krav, som at kunne vælge en bund og top til deres cupcake, og ydermere lave et login, hvor kunden kan gemme deres betalinger. Derudover skal admin kunne administrere kundens konto, ved at kunne slette kontoen eller fjerne en ordreliste.

Teknologivalg:

I projektet har vi anvendt en masse forskellige teknologier, men vi har primært brugt Java til at udarbejde projektet. Vi har brugt IntelliJ version 2021.3.2 (Ultimate Edition) og vi har sat projektet op med JDK 11.012. Udover Java har vi også haft en lokal Tomcat server til at køre vores web-applikation. Vi har kørt med Tomcat versionen 9.0.60. Vi har også været nødt til at gemme vores data, og til det har vi brugt en database, som vi har sat op med mysql. Vi har arbejdet i versionen 8.0.28 af mysql. Vi har brugt Maven til at hente nogle forskellige dependencies, som har hjulpet os til at udvikle projektet. Versionen af Maven er 4.0.0. Vi har bla. brugt javax.servlet version 4.0.1 og junit til at teste nogle af metoderne.

Krav

Vi har til dette projekt fået adskillige opgavekrav, som, hvis de blev opfyldt ville skabe en fungerende cupcake hjemmeside med mulighed for registrering, indlogning, bestilling, betaling samt ordreoversigt for både administrator og kunder.
idéen med kravene og håbet for virksomheden var at streamline bestillingsprocessen samt lave en overskuelig hjemmeside.

Ydermere fik vi udleveret disse user stories, som skulle opfyldes:

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 ordrelinier i en indkøbskurv, så jeg kan se den samlede pris.

US-5: Som kunde eller administrator kan jeg logge på systemet med email og kodeord. Når jeg er logget på, skal jeg kunne se min email på hver side (evt. i topmenuen, som vist på mockup'en).

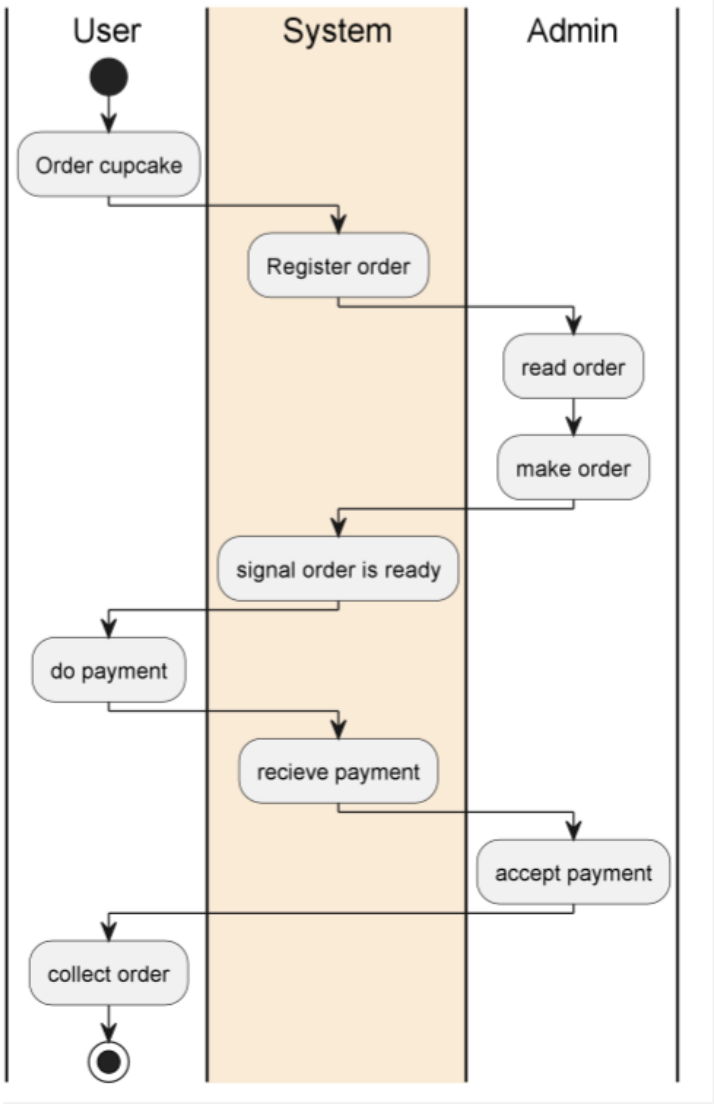
US-6: Som administrator kan jeg se alle ordrer i systemet, så jeg kan se hvad der er blevet bestilt.

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.

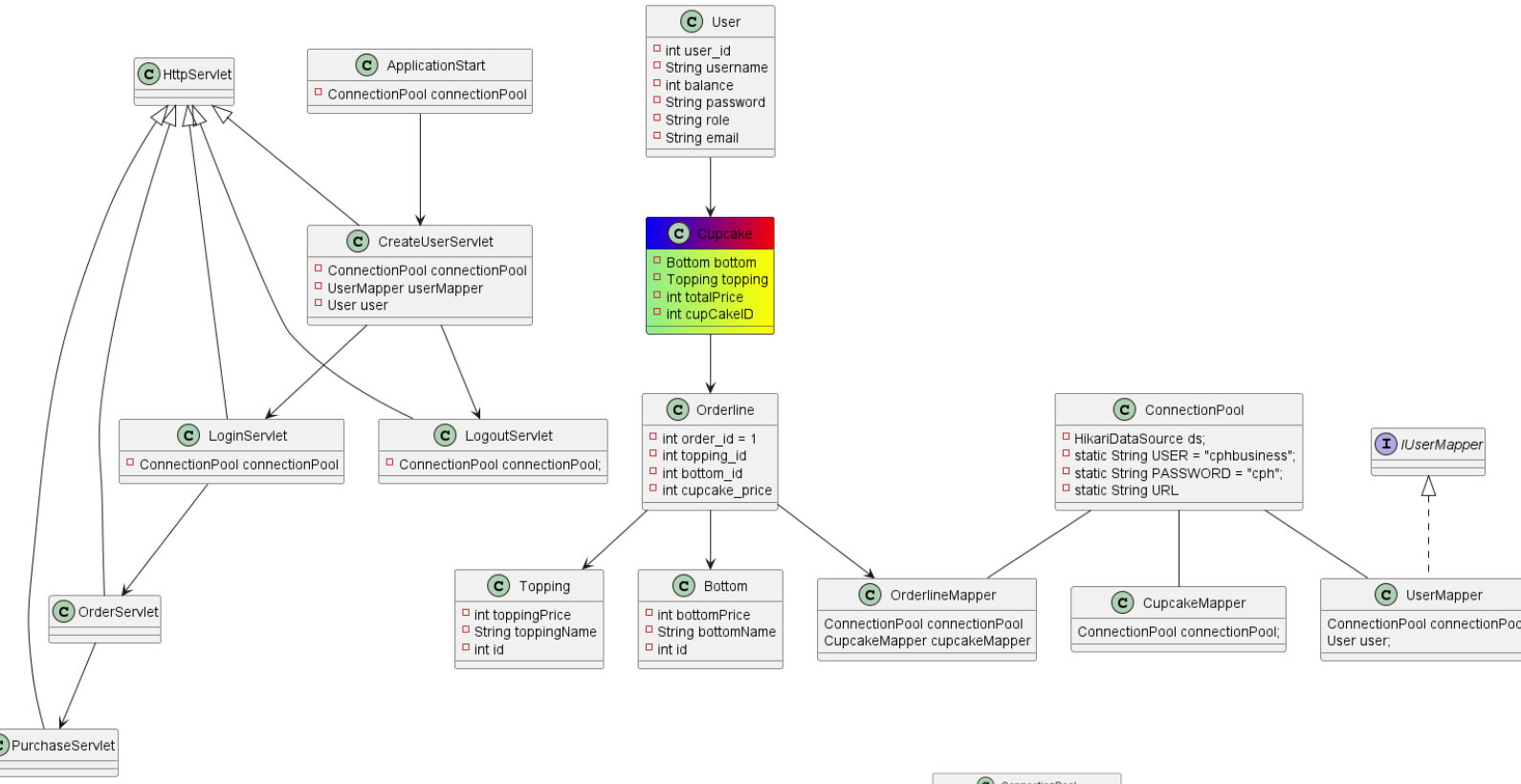
Aktivitetsdiagram

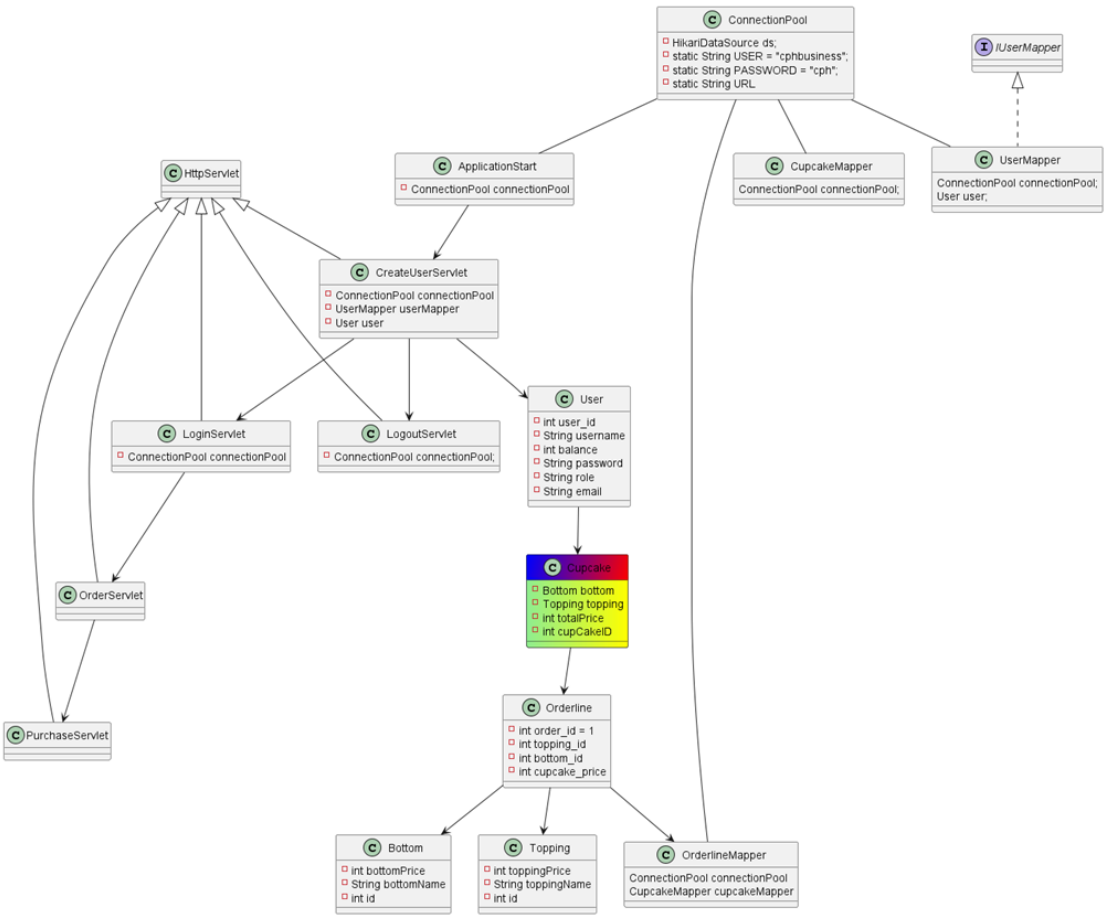


På diagrammet har vi forsøgt at illustrere hvordan en ordinær bestilling på vores hjemmeside ville have været. I retrospekt ville vi nok have gjort sådan at betaling ville ske umiddelbart efter at man har bestilt sine cupcakes, da dette fjerner chancen for at lave en ordrer forgæves.

Domæne model og ER diagram

Domæne diagram:





EER diagram:

EER diagrammet består af 5 enheder: User, order, orderline, bottom, topping.

Man kan se på diagrammet at der er et én til mange forhold mellem user og ordrer, dette er fordi vi tænker at en user godt kan være i stand til at have flere bestillinger på samme tid, men en ordrer kan kun have en user.

På diagrammet kan vi også se en orderline, hvilket er stedet vi samler selve cupcaken ved hjælp af en bottom og en topping. Forholdet mellem alle disse er en 1:1, da hver cupcake kun kan have en bottom og en topping.

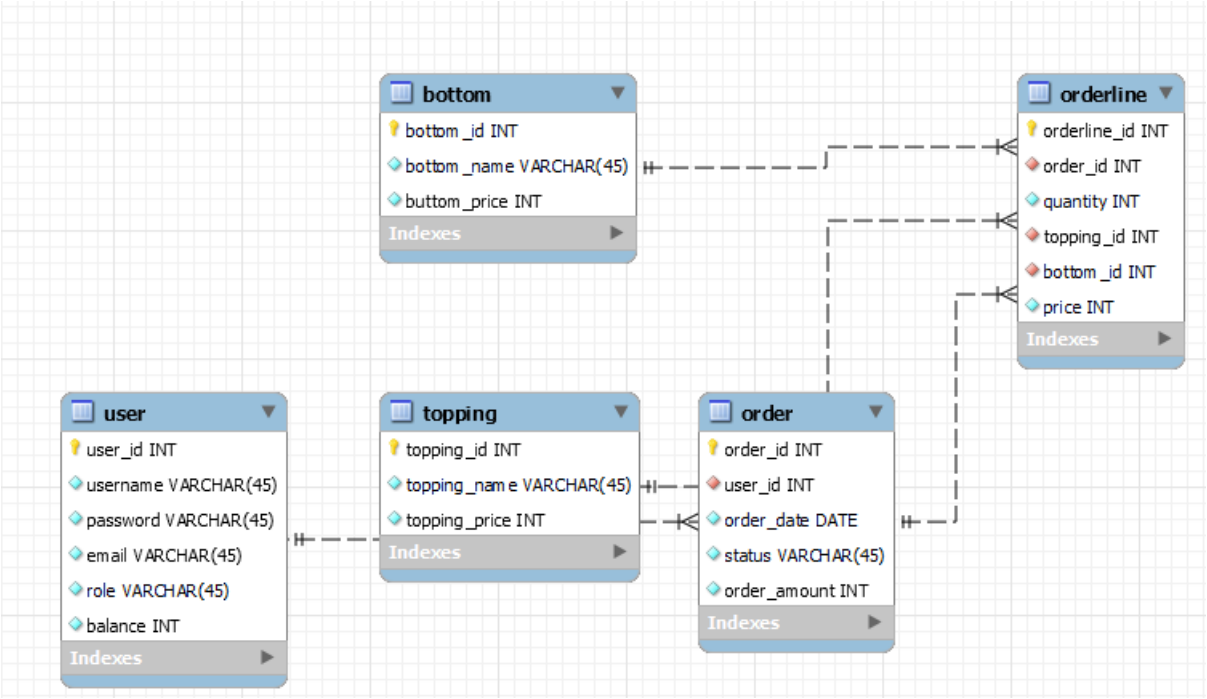
Vi har forsøgt at lave vores EER diagram med de forskellige normalformer in mente.

1NF er opfyldt, da ingen af kolonnerne gentager en anden kolonnes værdi, altså vi har ikke navn1, navn2 eller navn3 til at stå osv

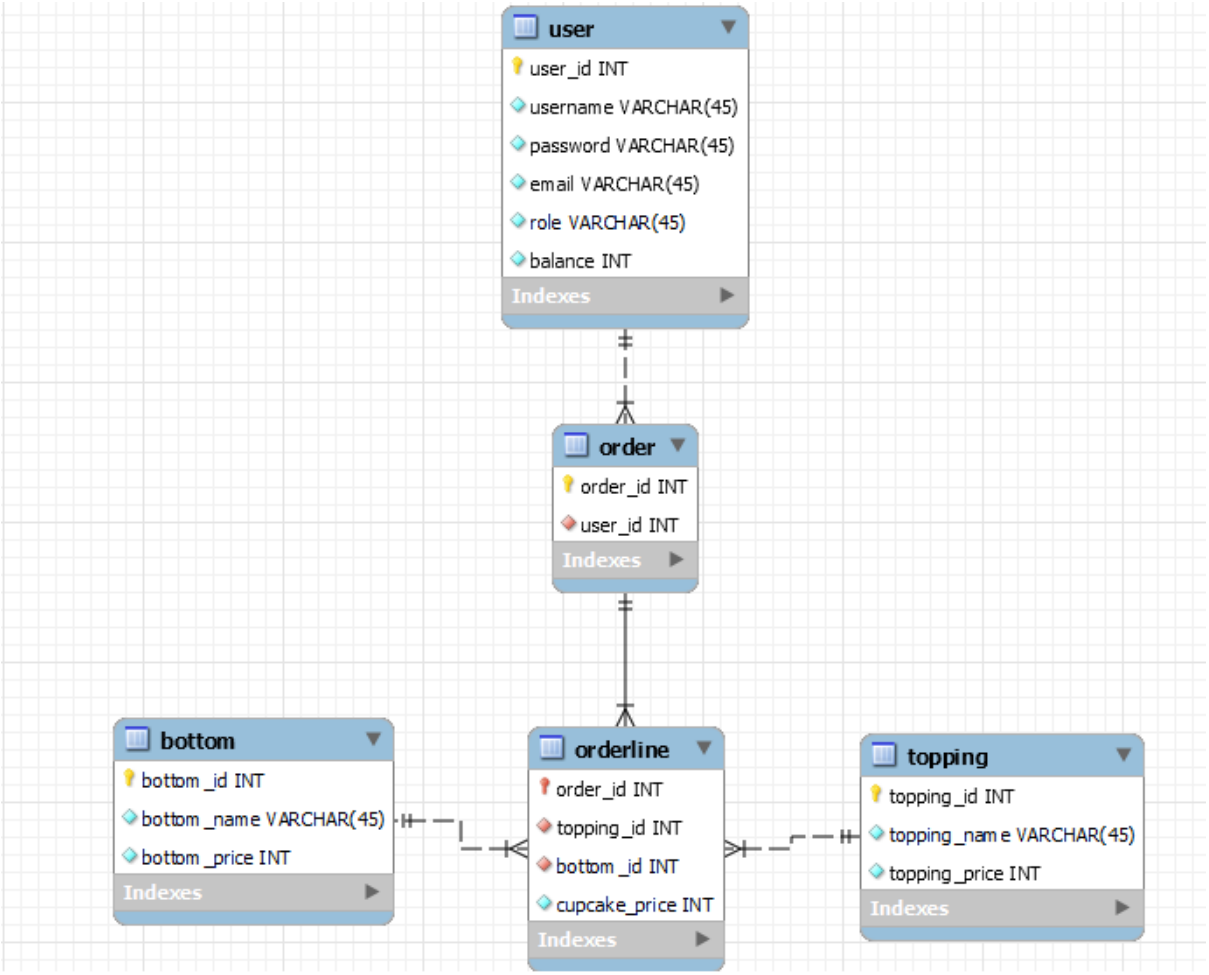
2NF er opfyldt. da hver kolonne kun har en unik nøgle, som henviser til den. I vores tilfælde er det oftest gjort med et _id

3NF har vi forsøgt at opfylde, da felter der er afhængige af hinanden er blevet spredt ud til deres egne kolonner og på den måde undgår vi redundans og skaber bedre overblik.

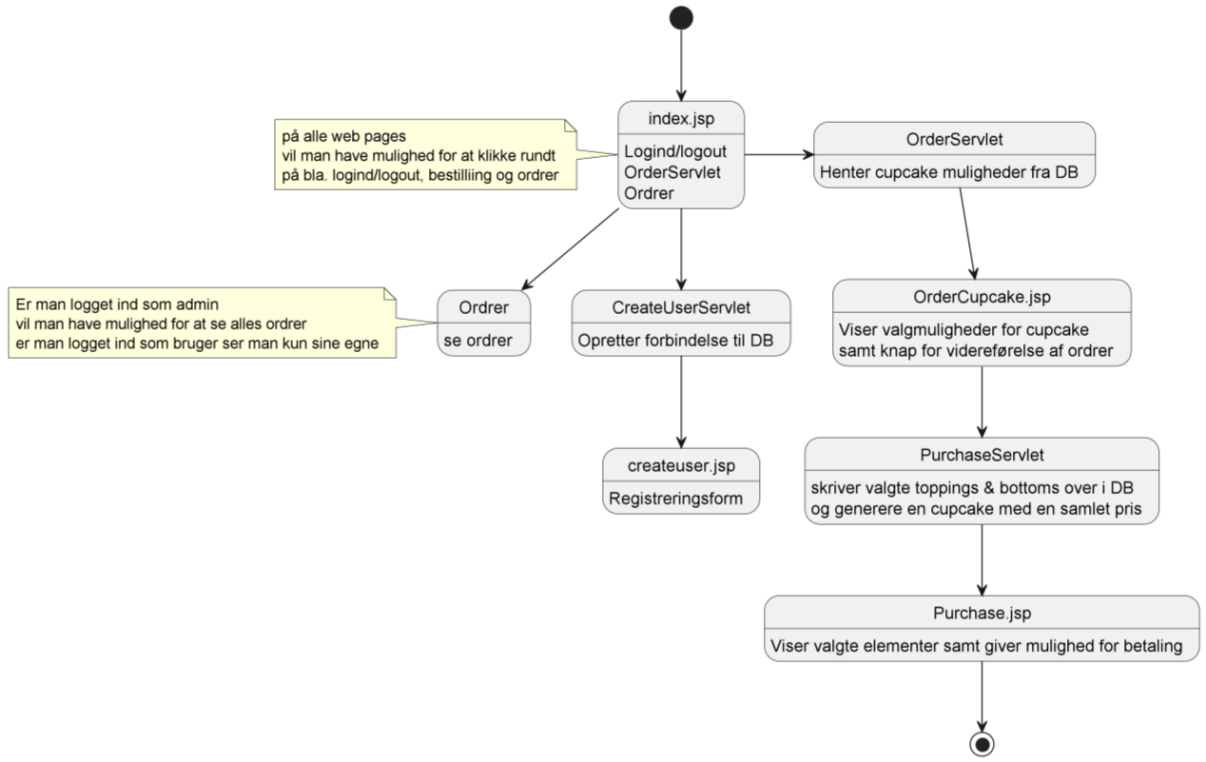
EER FØR (19-04-2022 EER Tabel under).:



EER EFTER:


user
user_id INT
username VARCHAR(45)
password VARCHAR(45)
email VARCHAR(45)
role VARCHAR(45)
balance INT
Indexes
order
order_id INT
user_id INT
Indexes
bottom
bottom_id INT
bottom_name VARCHAR(45)
bottom_price INT
Indexes
orderline
order_id INT
topping_id INT
bottom_id INT
cupcake_price INT
Indexes
topping
topping_id INT
topping_name VARCHAR(45)
topping_price INT
Indexes

Navigationsdiagram



Diagrammet her illustrere navigationen gennem vores program, som også vist på diagrammet så har vi en navigations bar øverst på hjemmeside, hvor man har mulighed for at tilgå bestilling, ordrer og login/logout. Ordrer siden vil ændre sig alt efter hvem der er er logget ind. Er du logget ind på en admin profil, vil du have mulighed for at se alle ordrer i systemet, hvorimod, hvis du er logget ind som en normal bruger vil du kun have mulighed for at se dine egne bestillinger.

Særlige forhold

Vores session starter når man logger ind, og når man logger ud har vi brugt session.invalidate() til at slutte vores session. Vi har bla. gemt brugerens informationer i vores session, så man kan hente brugerens data til hver en tid. Vi har f.eks. behov for at vide hvad brugerens email, role og balance er. Vi har også gemt alle brugerne i en Arraylist, som bliver gemt i vores session, så man har mulighed for at se en liste af kunderne som admin. Derudover gemmer vi også vores cupcake toppings og bottoms i vores session, så man har adgang til dem på vores bestillingsside.

I vores mappers har vi anvendt den database exception, som vi har fået udleveret, så vi ved når der opstår et problem og hvad grunden til det er.

Vi har ikke lavet en validering af brugerinput, der hvor man opretter en bruger, og det skyldes prioritering. Men hvis vi skulle have lavet det, så kunne vi have tjekket om brugerens password indeholdt store/små bogstaver, tal eller tegn som f.eks. !@$.

Vi har to brugertyper, en admin og en customer. Når man opretter en bruger bliver man automatisk tildelt rollen customer, og via databasen kan man ændre den til admin, så man får adgang til nogle features, som den almindelige customer ikke har.

Status på implementation

Vi startede med at sørge for at man kunne logge ind og oprette en bruger. Vi brugte vores Usermapper til at lave en login og createuser funktion, hvor vi har brugt SELECT og insert til at opdatere databasen. Disse metoder bliver så brugt i vores login- og createuserservlet, hvor vi gemmer dataene i vores session. Dernæst implementerede vi, at når man var logget ind, så fik man adgang til vores cupcakes, og når man er logget ind, så kan man også se ens email og balance i navigationsbaren. Som admin har man også en fane, der hedder customers, hvor man kan se en liste af alle customers. Ideen var at man skulle kunne se kundens ordre, men så langt nåede vi ikke.

Det som vi ville ønske, at vi havde nået at implementere er, at man kunne se de cupcakes, som man har valgt og betale for dem med sin balance. I vores orderservlet kan man vælge toppings og bottoms i en dropdown menu, men når man går videre for at bestille dem, så bliver de ikke gemt.

Process

Inden vi gik i gang, havde vi aftalt at fordele os ud på forskellige use-cases, som vi hver især ville sidde at arbejde med. Stødte vi på problemer havde vi aftalt at vi ville rådføre os med hinanden, før vi spurgte en lærer om hjælp.

Arbejdet kom til at forløbe som forventet og aftalt. Vi ville dog nok forsøge at løse problemerne selv i for lang tid, i stedet for at spørge om hjælp, hvilket resulterede i at nogle opgaver tog lang tid at løse og at der ikke var meget fremskridt hver dag.
Processen har givet os mere kendskab til kode-relateret gruppearbejde(branches og arbejdsfordeling), udover dette har vi uddybet vores forståelse for databaser, HTML, og web modellering.