Datamatiker 2. sem
Gruppe 2 Lyngby
21/04/2022

CUPCAKE
PROJEKT

Navn	Github	Mail
Andreas Fritzbøger	Wolfgang1235	cph-af167@cphbusiness.dk
Owais Dashti	OwaisAD	cph-od42@cphbusiness.dk
Thomas Fritzbøger	thomasfritzboger	cph-tf67@cphbusiness.dk

Indholdsfortegnelse

Indledning

Vi har fået til opgave at lave en hjemmeside for et cupcake bageri ved navn Olsker Cupcakes. På hjemmesiden skal Olsker Cupcake's kunder kunne logge ind, og bestille cupcakes med selvvalgt top og bund, samt antal, som de ønsker at afhente på bageriet. Så snart en ordre er placeret vil Olsker være i stand til at kigge på ordren og begynde at bage de ønskede cupcakes, der er bestilt af deres kunder.

Baggrund

Olsker Cupcakes er startet af et par hipstere fra København. De har deres bageri på Bornholm og deres fokus er økologisk kvalitet. Deres ønske er at tillade kunderne selv at designe cupcakes efter deres smag. Olsker Cupcakes ønsker man skal kunne bestille cupcakes, hvis man er oprettet som kunde, via deres nye webshop. Efter bestillingen er lavet, indsættes og gemmes data i en database, så bageriet kan finde ordren frem og påbegynde deres arbejde.

Teknologivalg

Vi har benyttes os af følgende teknologier.

Programmering	***Brugte software***	***Frameworks***
Java (17.0.2) MySQL (DB) (8.0.27) HTML5 og CSS3 JSP (3.0)	IntelliJ IDEA (2021.3.2) MySQL Workbench (8.0 CE) Draw.io (16.5.1) Figma Safari, Firefox, Google Chrome og Edge	MySQL connector (8.0.28) JSTL (1.2.7) Bootstrap (5.0)

Kundens krav

Der er blevet afholdt et kundemøde, som har udmøntet sig i en række user-stories. Der er 9 i alt, hvoraf de 6 første er obligatoriske, mens de resterende er mulige tilvalg.
Vi har valgt en løsning, hvor vi forudsætter, at en kunde kun kan købe cupcakes op til den balance, man har på sin konto. Dvs. man altid er godkendt til køb for gennemførte ordrer. Det gav derfor ikke mening, at der i vores kontekst skulle være mulighed for at annullere en ordre, fordi den ikke var betalt.

I punktet status på implementation vil vi fremføre forslag til forbedringer, som Olsker Cupcakes kunne ønske at implementere senere.

Herunder er de 8 gennemførte user-stories og hvordan vi har implementeret dem:

- *"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."*

De 5 typer af bund og de 9 typer af topping er placeret i en dropdown menu, således, at der ikke ved en fejl kan vælges varianter, som ikke produceres. Der er i udgangspunktet mulighed for at vælge op til 15 stk af samme type, men når man checker sin ordre i kurven, er der mulighed for at tilføje flere alt efter ens maksimale kredit
Kurven kan opdateres inden den placeres, hvis man ønsker at tilføje eller fjerne cupcakes.
Der kan kun placeres ordrer, som der er saldo for på kundens konto.

- *"US-2 2: Som kunde kan jeg oprette en konto/profil for at kunne betale og gemme en ordre."*

I den login modal/form, der popper op på forsiden, er der en knap til endnu en modal, hvor man kan oprette sig som kunde, hvis man ikke allerede har en konto. Mht. begrænsningen ift. betaling af en ordre, se evt. under user-story 1. Der er kun

mulighed at oprette sig som kunde via hjemmesiden, en administrator skal oprettes manuelt gennem MySQL databasen.

- *"US-3: Som administrator kan jeg indsætte beløb på en kundes konto direkte i MySql, så en kunde kan betale for sine ordrer."*

Når en administrator er logget ind, er der en knap i navigationsbaren, som fører til en jsp-side, hvor det er muligt at tilføje kredit til de eksisterende kunders konti. Denne del ligger uden for vores kundes ønske, men vi valgte at tilføje denne funktion, så man ikke kun skulle kunne opdatere Olsker kunders kredit via MySQL workbench. Hvis der indsættes beløb på en konto, som ikke eksisterer, får man ikke en fejlbesked på nuværende tidspunkt, og man vender blot tilbage til listen af kunder uden andet opstår.
Betaling og indsættelse af kredit til ens balance som kunde er ikke beskrevet dybere, men skal foregå fysisk eller ved kontakt til Olsker Cupcakes evt. via servicen mobilepay, herefter skal administrator opdatere kundens saldo.

- *"US-4: Som kunde kan jeg se mine valgte ordrelinjer i en indkøbskurv, så jeg kan se den samlede pris."*

Når man er logget ind som kunde, kan man, når man står på "kurv-siden", se ens samlede ordre, som specificeres i de enkelte cupcake typer, herunder antal af hver type og den samlede pris for ordren.
Det er endvidere muligt at se den tilrådighedværende kredit, og hermed om man har råd til at placere ordren, eller om den skal tilpasses.
Under kundens profil, som kan tilgås gennem navigationsbaren, kan kunden desuden se en liste over tidligere ordre.

- *"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."*

Der er 2 mulige knapper til at logge ind med. En øverst i højre hjørne, og en nederst midt på forsiden. Den ene er navngivet "log ind" og den anden med "start din rejse". I

navigationsbaren vil kundens eller administratorens email fremgå på alle sider, såfremt de er logget ind.

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

Den første side man som administrator ser, efter man logger ind, er en liste over alle ordrer i systemet. Der er endvidere en mulighed for at se alle ordrelinjer (cupcake varianter og antal) for en specifik ordre, ved at trykke på knappen "Se ordre indhold" ud for ordren i tabellen.

- *"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."*

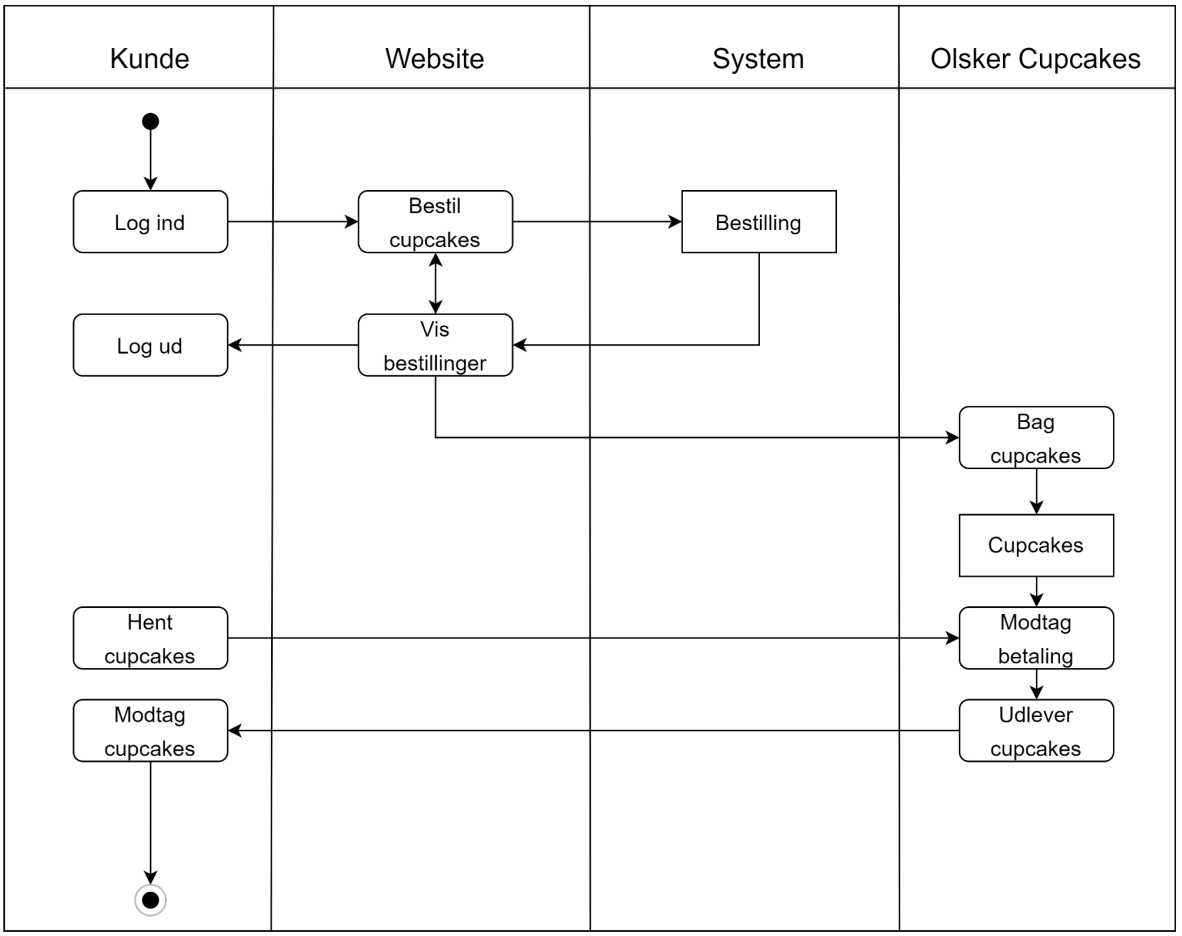
Efter en administrator er logget ind kan denne gennem navigationsbaren, navigere til "kunder" siden, der giver administratoren mulighed for at se alle kunder i databasen og deres balance/kredit. Det er endvidere muligt her, at tilføje yderligere kredit til brugeren - hvilket er gjort for at undgå at skulle gøre dette via MySQL.
Mht. ordrer beskrivelse, se ovenfor US-6.

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

Det er muligt at tilføje og fjerne cupcakes direkte i kurven, samt navigere tilbage til cupcakefactory jsp-siden for at tilføje nye typer cupcakes til ens kurv. Endvidere er det muligt via profilsiden at se tidligere placerede ordrer.

Diagrammer

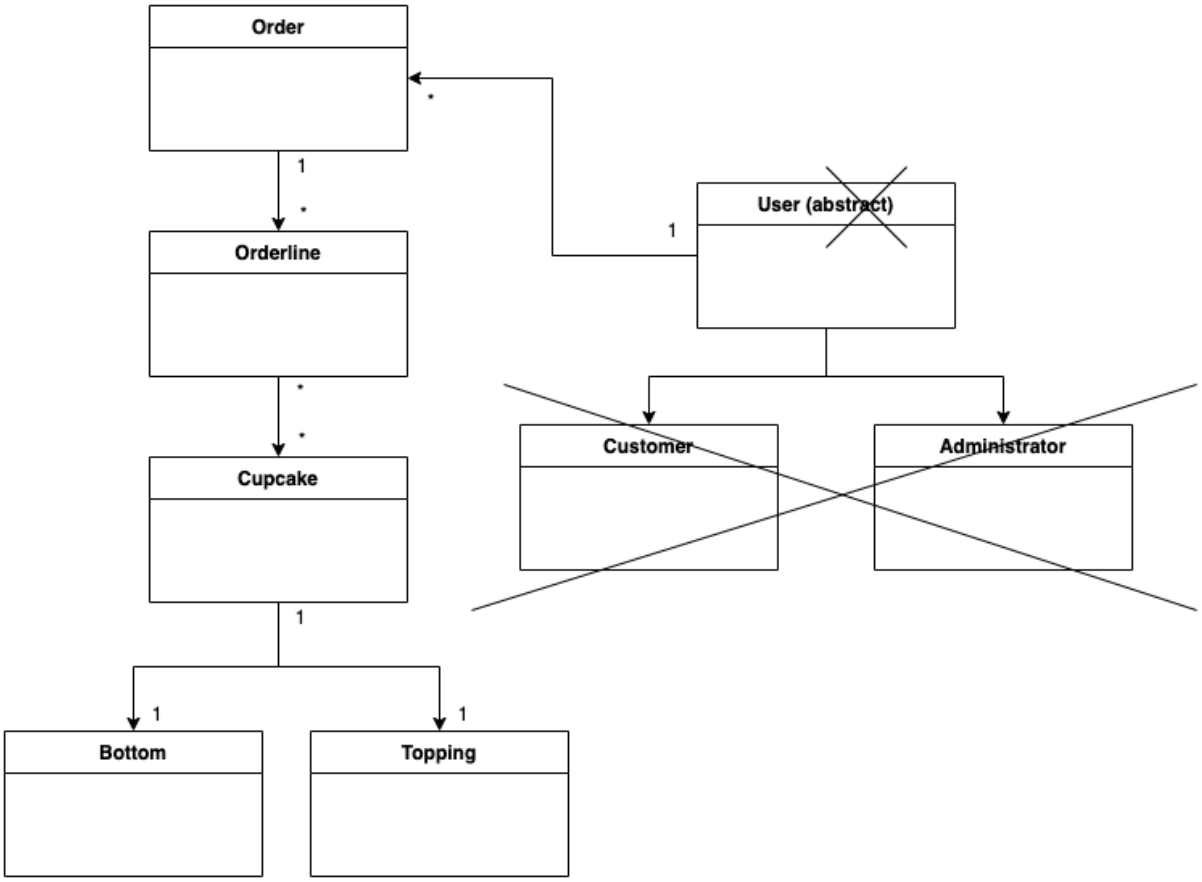
Aktivitetsdiagram



Når en kunde logger ind på Olsker Cupcakes' via forsiden, sendes kunden til cupcakefactory siden, hvor man her kan designe sine cupcakes, og angive hvor mange man vil have til bestilling. Bestillingen bliver gemt i en database, og kan hentes ind og blive fremvist af kunden og administratoren af Olsker Cupcakes. Kunden kan kun se sine egne bestillinger, der er blevet gemt, mens administratoren kan se en liste af alle kunders bestillinger. Kunden kan herefter vælge at fortsætte med at bestille flere cupcakes, se alle sine egne bestillinger, eller logge ud. Bagerne fra Olsker Cupcakes vil gennem administrator siden kunne finde kundens bestilling frem og bage de valgte cupcakes. Det er tiltænkt at kunden fysisk skal hente de bestilte cupcakes på bageriet. Så snart bagerne fra Olsker Cupcakes har modtaget betaling fra kunden, og ordren dukker op på deres ordreside, vil bagerne bage/pakke

de ønskede cupcakes og udlevere dem til kunden, når kunden kommer ind i bageriet.

Domænemodel

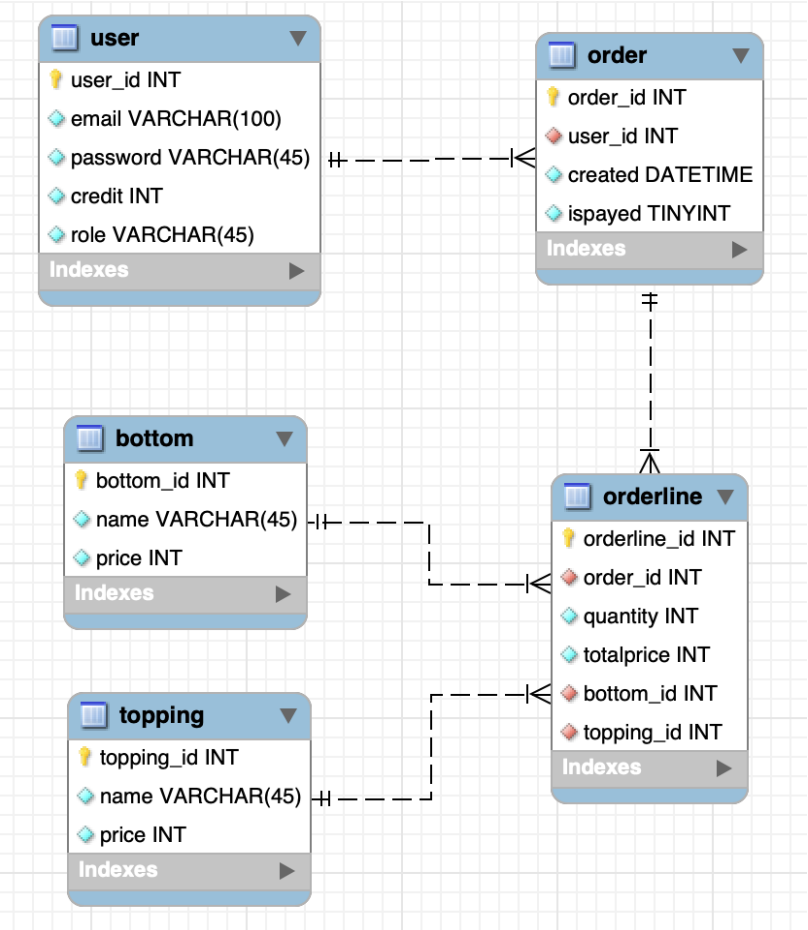


En af de første og primære modeller vi fik lavet var domænemodellen, som set ovenfor. Denne er lavet for at give os et overblik over hvad systemet skal håndtere og herunder hvordan hierarkiet kommer til at se ud. Det er vigtigt at repræsentere de forskellige enheder i systemet og hvordan de er involveret ift. hinanden. Denne domænemodel lavede vi inden vi påbegyndte kodningen af selveste programmet. Vi kan bruge denne model til at reflektere og få et overblik over den verden som programmet skal “leve” i, således at både kunden og udviklerens forventninger stemmer overens.

Vores domænemodel viser at man i programmet kan have to mulige brugere, nemlig Customers og Administrators.

Udover de 2 brugertyper, skal man kunne placere nogle ordrer, som består af ordrelinjer, som igen består af et antal cupcakes, der har en slags bund og en slags topping.

ER-diagram

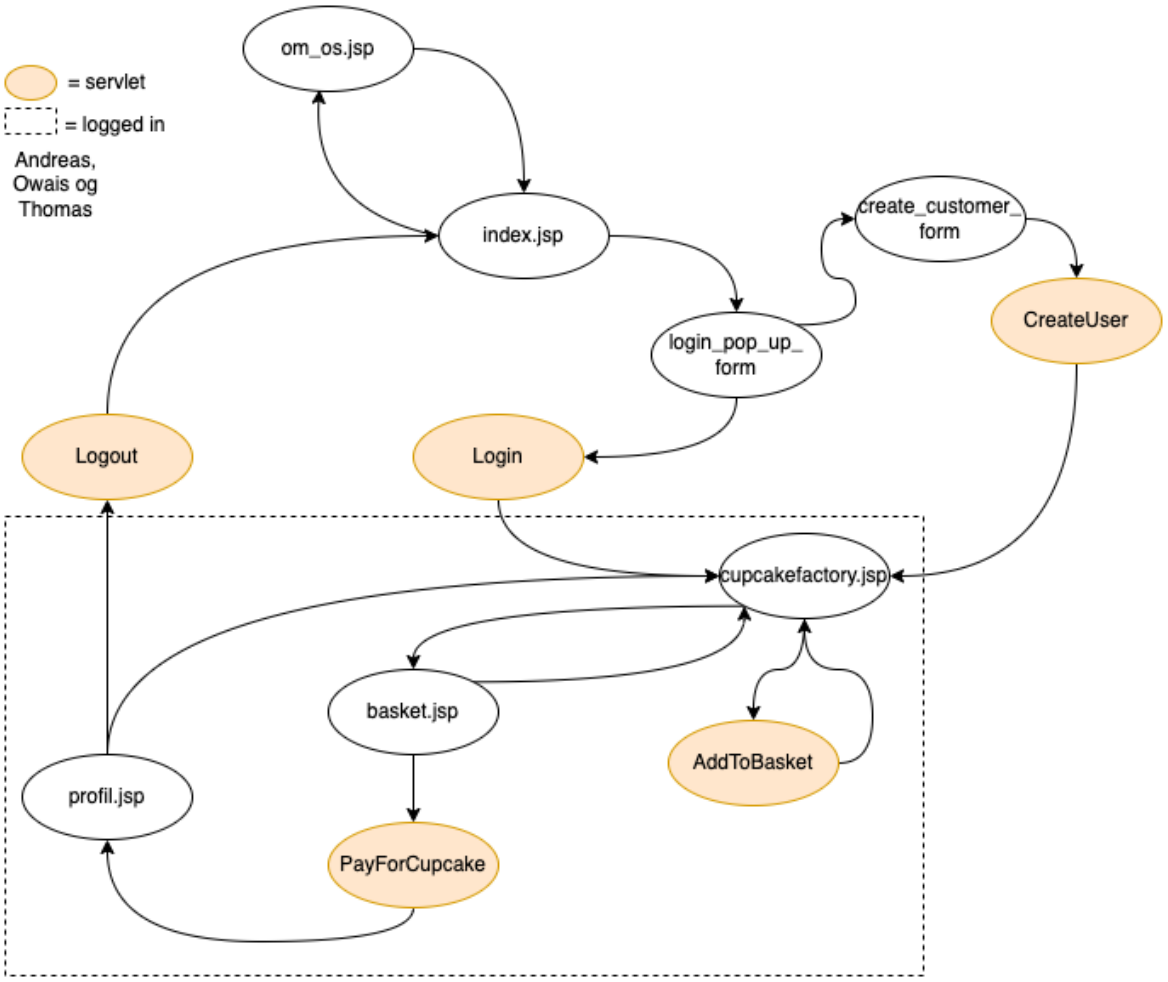


Vi designede vores relationelle database og nåede frem til ovenstående skabelon. Vi har forsøgt at følge de 3 primære normalformer, for at eliminere redundans i tabellerne, da vi ønsker at opnå en effektiv, overskuelig og hensigtsmæssig struktur, således at vi kan udføre effektive og korte forespørgsler gennem vores datamappere. Vi opnåede ikke helt at være i 3. normalform da vores 'User' tabel reelt set kan splittes, således at man kunne have en separat tabel for roller. Vi har haft brug for at samle dataen mellem forskellige tabeller, hvor vi dertil fik oprettet nogle views, hvori vi joiner forskellige tabeller. Dette simplificerede ligeledes vores forespørgsler til en række af vores tabeller på hjemmesiden. En anden pointering er benyttelsen af 'isPayed' attributen under 'order' tabellen. Denne kunne undlades i dette projekt, da vi ikke når at implementere en særlig købsfunktion som reelt hæver penge i virkeligheden. Vi har i flere tabeller med constraints. Alle vores tabeller indeholder variabler hvor de alle er 'not null'. Et andet eksempel med constraints er 'email' i 'user'-tabellen hvor denne værdi er sat til at være 'unique'. Mht. relationer

mellem tabellerne, sker det kun gennem fremmednøgler, dvs. at tabellerne blot afhænger af hinandens primærnøgler.

Navigationsdiagrammer

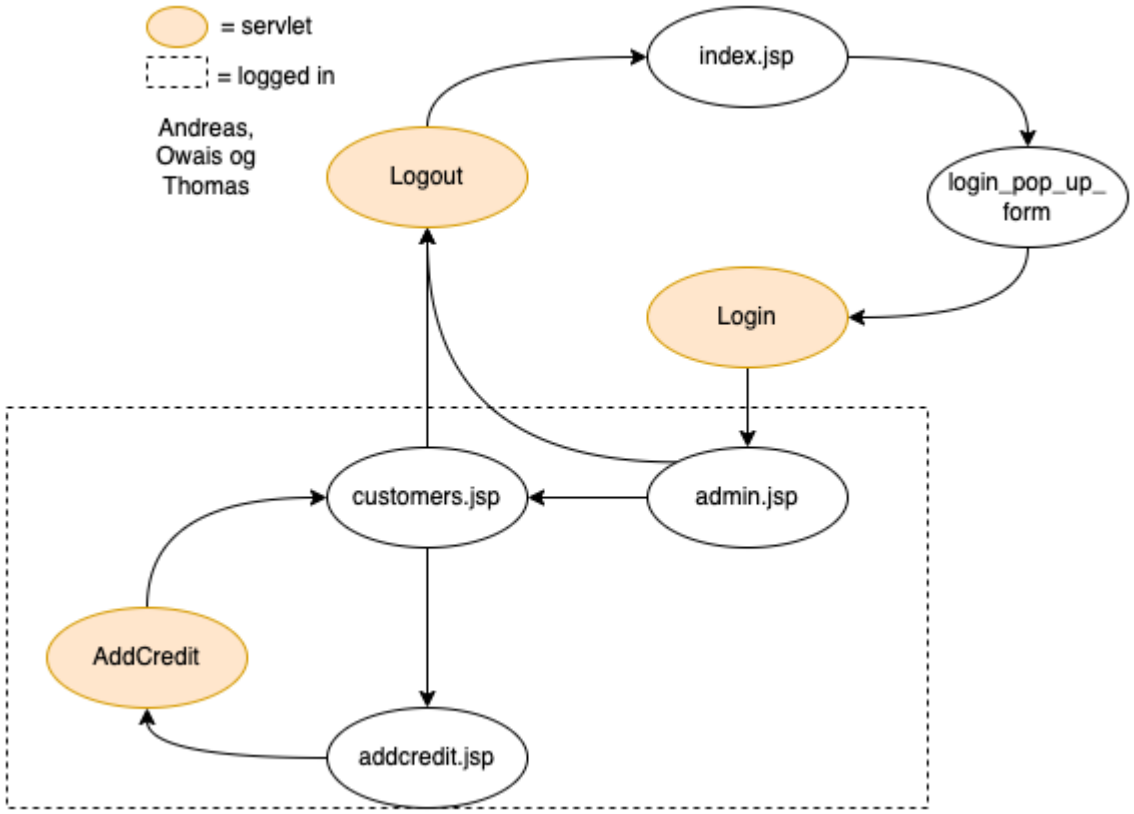
Navigationsdiagrammerne giver os, som udviklere et overblik over flowet i vores applikation. Vi har nogle sider, som kunden ser, og nogle som administratoren i systemet ser. Der er ligeledes nogle servlets bag facaden som gemmer og flytter data rundt mellem siderne, således man kan hive fat i dem senere hen. Vores udkast af de følgende navigationsdiagrammer var foreløbige på daværende tidspunkt, og vi måtte senere tilføje flere servlets alt efter hvor det blev nødvendigt. Dette beskrives yderligere under punktet status på implementation.



Tiltænkt navigationsdiagram for bruger-rollen startende på index.jsp (model 1)

Vores navigationsdiagram set foroven er tiltænkt en kunde i systemet. For en kunde vil navigationen se ud som følgende (underbygget af model 1):

1. Kunden starter på forsiden, altså index.jsp siden. Her kan kunden navigere mellem forsiden og om os siden i navigationsbaren.
2. Kunden kan logge ind eller oprette sig som kunde på de primære sider via en pop up modal/form som ligeledes kan tilgås gennem en knap i navigationsbaren. Er kunden ikke medlem kan en ny pop up form tilgås gennem den tidligere åbne form, hvorunder kunden kan oprette sig.
3. Er kunden allerede oprettet sendes denne videre til cupcake factory siden. Hvis der er fejl i de indtastede oplysninger, sendes kunden til en fejlside, hvorefter kunden kan vende tilbage til forsiden eller klikke på log ind knappen i navigationsbaren, for at forsøge at logge ind igen. Opretter kunden sig som medlem succesfuldt sendes kunden ligeledes, efter oprettelsen direkte videre til ‘cupcakefactory’ siden.
4. Herfra kan kunden nu tilføje cupcakes til kurven eller tilgå sin kurv eller profil side i navigationsbaren.
5. I kurven kan kunden se evt. tilføjede cupcakes. Er kurven tom kan kunden vende tilbage til ‘cupcakefactory’ eller logge ud. Vil kunden udføre ordren kan dette gøres, hvorefter kunden automatisk sendes til profilsiden, hvor ordren kan findes, hvis den er gået igennem.
6. Kunden vil til enhver tid kunne se tidligere ordre under profilsiden og logge ud i navigationsbaren.



Tiltænkt navigationsdiagram for administrator-rollen startende på index.jsp (model 2)

Vores navigationsdiagram set foroven er tiltænkt en administrator i systemet. For en administrator vil navigationen se ud som følgende (underbygget af model 2):

1. Administratoren starter på forsiden, altså index.jsp siden. Her kan administratoren navigere mellem forsiden og om os siden i navigationsbaren.
2. Administratoren kan logge ind i via en pop up form.
3. Logger administratoren ind uden fejl i oplysningerne, sendes denne videre til admin siden som lister alle eksisterende ordre i systemet. Er der fejl i loginoplysningerne sendes administratoren til en fejlside hvorefter denne kan vende tilbage til forsiden og forsøge at logge ind igen.
4. I navigationsbaren kan administratoren navigere mellem en ordre-side og en kunde-side. Ordre-siden lister alle ordrerne i systemet. Kunde-siden fremviser en liste af eksisterende kunder i systemet.
5. Administratoren kan i kunde-siden tilføje kredit til en brugers balance. Herefter sendes administratoren tilbage til kunde-siden, hvor balancen automatisk ændres.

6. Administratoren vil til enhver tid kunne logge ud i navigationsbaren.

Mockups

Vi har designet mockups til programmet. Den kan findes under mappen /dokumentation/mockups i vores github repository og eller på følgende side: https://tinyurl.com/526ay4ux

Demonstration af resultat

Her er et link til demonstration af projektet. Der ageres først som kunde på siden og efterfølgende som administrator.
https://www.youtube.com/watch?v=j9loEFQMb3k

Særlige forhold

Topping og bund typer gemmes i sessionen, men kunne i nuværende opgave-kontekst gemmes i application scope, da det ikke er muligt at opdatere cupcake typer.

Usertype, kunde eller admin, gemmes i sessionen og ligeså med e-mail for den bruger, der er logget ind.

Vi har valgt ikke at kryptere password, det bliver på næste semester.

Ved log ind og opret bruger, bliver brugernavn valideret som e-mail, dvs det skal indeholde @, password i log ind vises ikke direkte. I opret bruger bliver password dels verificeret som værende det samme i begge input felter, ligesom længden på password bliver testet, der er pålagt meget simple krav til password.

Der er i denne løsning ikke lagt begrænsning på antallet af forsøg på log ind.

Hvis man forsøger at oprette en bruger med et username, der allerede er i brug, får man besked om dette og føres til error.jsp.

Vi benytter os af prepared statements i datamapperne, således at vi undgår SQL injektioner.

Når man logger ud, stopper sessionen, men man kan stadig gå tilbage og se de besøgte sider, man kan dog ikke lave inputs (eks bestille cupcakes eller tilføje kredit). Man vil kunne se at man igen skal logge ind.

De steder, hvor der var risiko for runtime errors (eksempelvis når kurven opdateres), har vi gjort input required, for at undgå nedbrud.

Status på implementation

Der er ikke lavet unit test til denne opgave, da det ikke var en del af projektet.

Der er ikke lavet java doc, da dette ikke var en del af projektet, men der er tilføjet enkelte kommentarer i filerne, hvor vi fandt, at det gav mening.

Vi har afveget fra domænemodellen ved ikke at lave en abstrakt klasse, men isf givet user en klassevariabel “role”l i form af en String.

Boolean i ordre tabellen bliver ikke brugt, da vi altid har betalte ordrer, men det vil give mening i en videreudviklet udgave at bruge boolean som isDelivered isf isPayed, for at beskrive om den er afhentet eller ej.

Fra cupcake.jsp kan man føres via ProfilNavigation som er en ny servlet, der henter data om alle kundens ordre og videre til profil.jsp, der viser alle kundens ordrer

Fra basket.jsp føres man via UpdateInBasket som er en ny servlet, der opdaterer kurven, hvis der tilføjes eller fjernes fra denne

Fra admin.jsp kan man føres via OrderDescription, som er en ny servlet, der henter data vedrørende den enkelte ordre

Efter login, hvis man er administrator føres man først videre til AdminNavigation servletten, som er en ekstra ift navigationsdiagrammet.

Man kan diskutere om vi skulle have gjort det muligt, kun at se en specifik kundes ordrer, dette er ikke implementeret, da sorteringen i vores opgave kun er på ordre_id.

Af grunde vi ikke har kunne fundet ud af endnu, mister vi footeren samt højre box, hvis man som kunde går tilbage fra kurven til cupcake factory. Dette problem opstår kun hvis man bruger Chrome eller Edge, men ikke Safari eller Firefox. Vi har forsøgt at debugge, og fejlen skyldes muligvis et missing tag, som vi har ledt efter uden held.

Fremtidige forbedringsmuligheder:

- Administrator skal kunne oprette/nedlægge bund og topping muligheder direkte ved login isf at skulle gøre det i mySql. Der skal ligeledes være mulighed for at ændre priser
- Administrator skal have mulighed for at lave tilbudspakker.
- Ordrelisten på administrator-siden, skal tilføjes en knap, hvor man kan markere ordren, som værende afhentet, således, at det kun er betalte uafhentede ordrer man ser. Der kan/bør tilføjes endnu en knap hvor man kan se alle ordrer, både de afhentede og de uafhentede, der kunne endvidere tilføjes en filtrerings mulighed.
- Man kunne tilføje en bruger/admin en boolean, således at kontoen låses, hvis der laves et antal forsøg på login, som ikke bruger rigtige password.
- I samme forbindelse kunne man gøre det muligt at reset sit password.
- Administrator skal have en fejlbesked, hvis der forsøges indsat penge på en ikke-eksisterende konto.
- Der kunne laves en knap til transaktioner, evt med filtreringsmuligheder.
- Det skal være muligt at angive et tidspunkt for, hvornår en ordre er klar til afhentning.

- Ordreinholde kan fremvises som et popup i stedet for direkte over en tabel. Dette vil forbedre den nuværende løsning, da tabellen kan være lang og hver gang den benyttes vil den hoppe til top.

Proces

Der er et link til vores logbog i README.md, som beskriver hvad vi har arbejdet med de enkelte dage. Det har været lærerigt at arbejde på denne måde, og vi er enige om, at samarbejdet er gået rigtig fint.