

Indhold

Indledning:.....	2
Baggrund:	2
Teknologier:.....	2
Krav:.....	3
Aktivitets Diagram:	4
Domæne Model:.....	5
ER Diagram:	6
Navigations Diagram:	7
Særlige Forhold:.....	7
Proces:	8

Cupcake projekt

Af Mads Herlevsen og Casper Posche

Indledning:

Indledningen er ment til en anden 2. sem studerende som evt. ikke kender cupcake opgaven

Vi fik til opgave at lave en hjemmeside til et bageri hvor man skulle kunne skræddersy og bestille cupcakes og hente dem i butikken, man skulle som admin kunne se ordre og kunder, og se om ordrende var betalt eller ej

Baggrund:

Virksomheden er et bageri der laver cupcakes

Man skal som kunde kunne skræddersy en cupcake hvor man vælger en bund og en top med individuel pris, som derefter skal ligges i en kurv hvor man kan se sine nuværende ordre og derefter betale og hente dem i butikken.

Derudover skal en administrator kunne logge ind og se alle ordre samt alle kunder og deres ordre.

Kunder skal kunne oprette et login med en e-mail så der evt., kan sendes en ordre bekræftelse når ordren er betalt/bestilt

Teknologier:

IntelliJ 2021.2.3 Ultimate Edition – brugt som IDE

JDBC – brugt til at implementere database brug

JSTL 1.2 – brugt til at køre html på Java via .jsp sider

HikariCP 5.0.1 – brugt til håndtering af flere forbindelser via connectionpools

Lombok 1.18.22 – brugt til at auto genere getters og setters

MySQL 8.0.28 – brugt som database i projektet

Java servlet 4.0.1 – brugt til at håndtere http request og response på en server

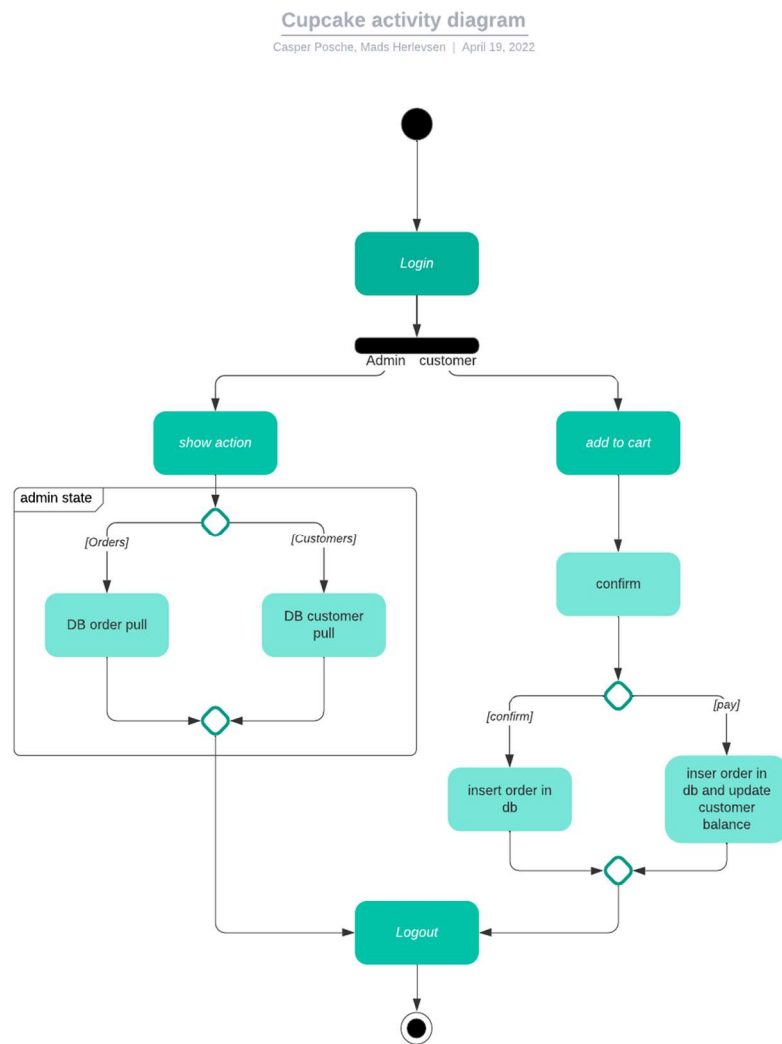
Apache Tomcat server 9.0.60 – brugt som server til at køre hjemmesiden

Krav:

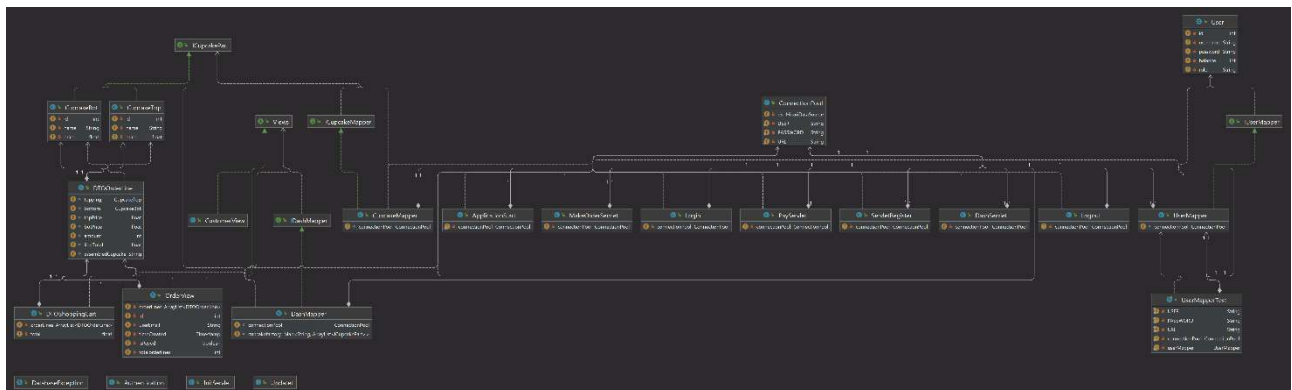
Firmaets håb med vores produkt er at mindske trafik i deres butik, og gøre det nemt at bestille og betale online, så man bare skal komme og hente cupcakesne.

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.	Denne funktion er implementeret 100% vi har dog ekstra funktionalitet som ikke virker 100%
US-2 Som kunde kan jeg oprette en konto/profil for at kunne betale og gemme en ordre.	Denne funktion er implementeret 100% vi har dog ikke fået lavet en profil side hvor man kan se sine ordre
US-3: Som administrator kan jeg indsætte beløb på en kundes konto direkte i MySQL, så en kunde kan betale for sine ordrer.	Vi havde tænkt os at implementere at når kunden bestilte og betalte blev der automatisk trukket fra i databasen, men vi nåede det ikke grundet tidspres
US-4: Som kunde kan jeg se mine valgte ordrelinier i en indkøbskurv, så jeg kan se den samlede pris.	Denne funktion er 100% implementere
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).	Denne funktion er 100% implementere
US-6: Som administrator kan jeg se alle ordrer i systemet, så jeg kan se hvad der er blevet bestilt.	Denne funktion er 100% implementeret dog med manglende ekstra funktionalitet som vi havde udtænkt os
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.	Denne funktion er ikke implementeret endnu da vi ikke havde tid, men den kan nemt implementeres dag den ligner meget vores US-6 og vi har lavet det så det forgår på samme 'admin' side som hedder Dashboard
US-8: Som kunde kan jeg fjerne en ordre fra min indkøbskurv, så jeg kan justere min ordre.	Denne funktion er desværre ikke implementeret endnu
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.	Denne funktion er desværre ikke implementeret endnu

Aktivitets Diagram:

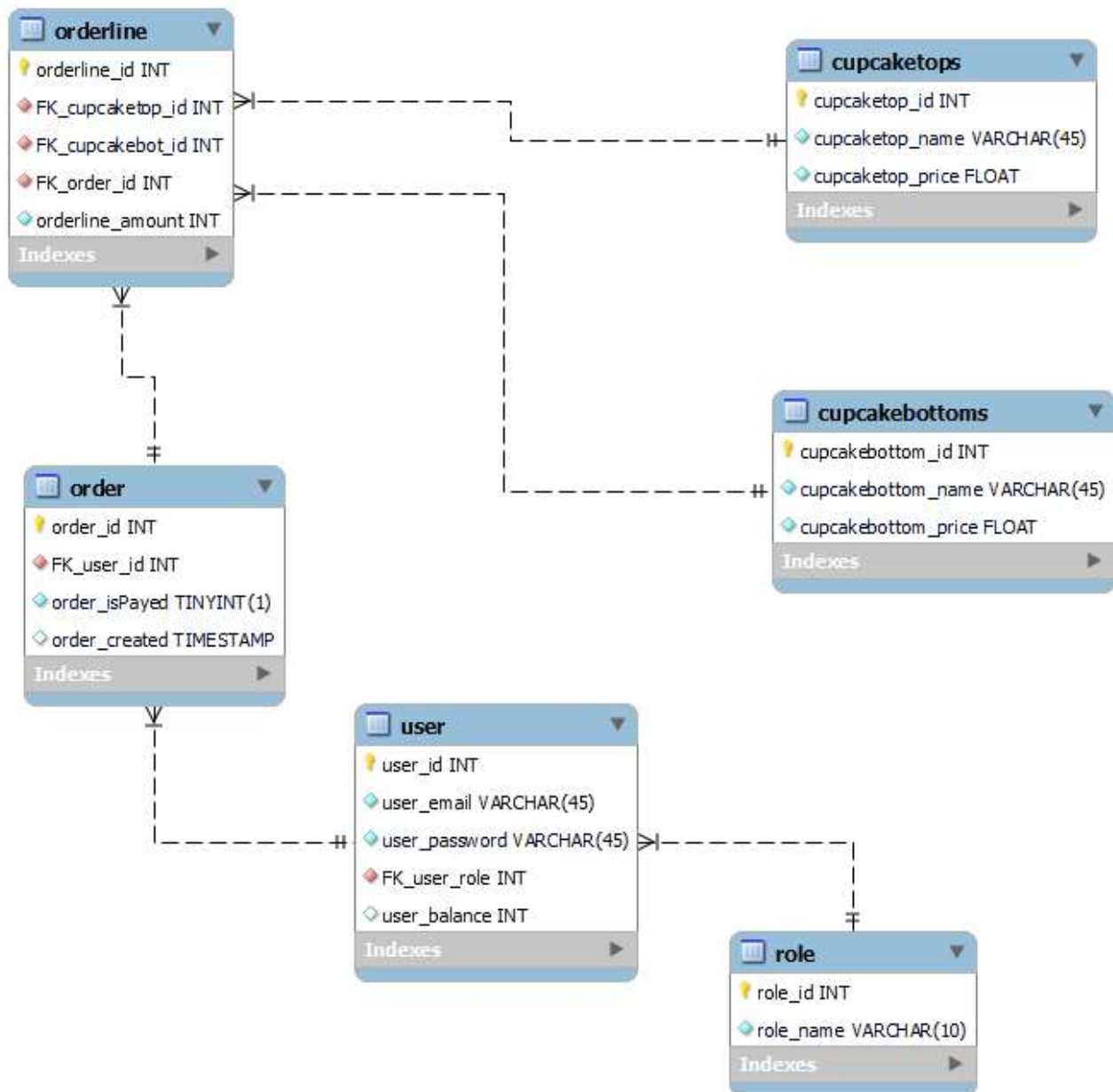


Domæne Model:

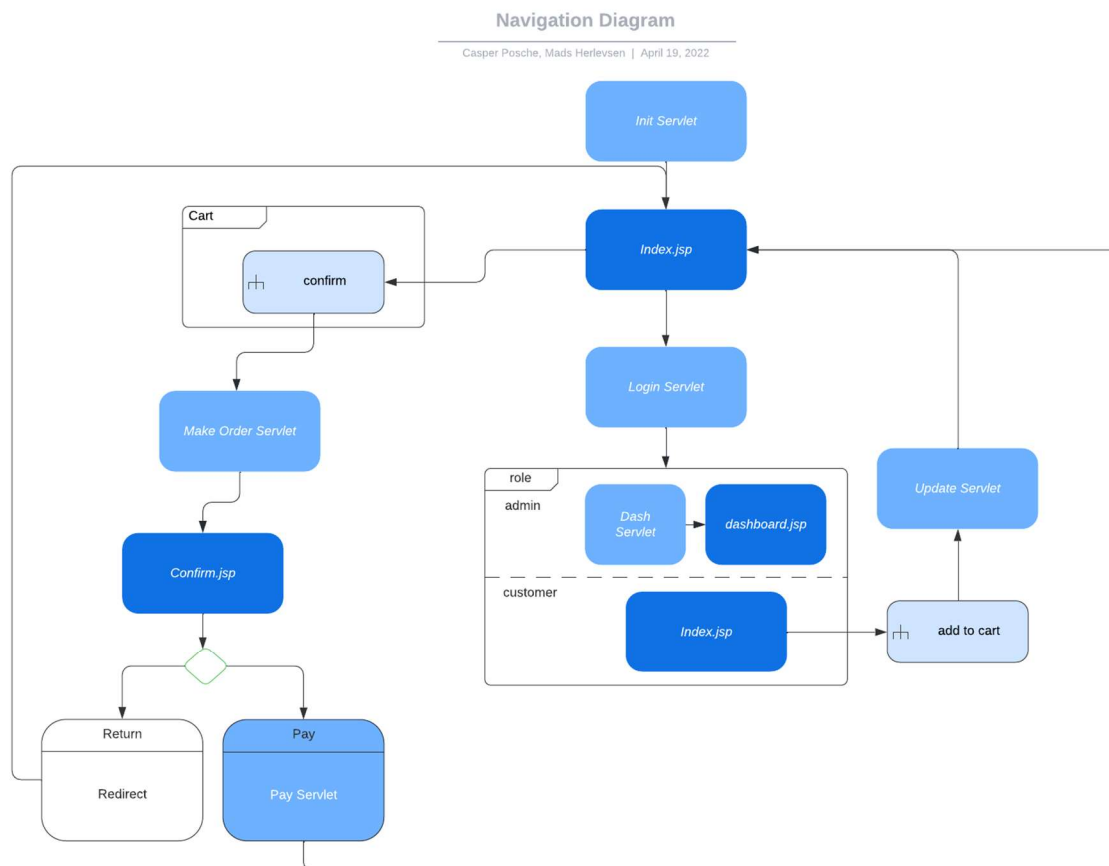


Med vores cupcakes har vi studset meget over hvordan de skulle relatere til hinanden, f.eks. om de skulle samles i en cupcake klasse, eller som vi endte med at gøre have en top og en bund klasse der først samles i en ordrelinje, og før vi rigtig begyndte, havde vi ikke et interface, men da vi valgte at ligge dem som et Map på applikations scopet, var det nemmere at bruge et interface, så man kunne 'switche' mellem dem, uden at lave for mange klasser.

ER Diagram:



Vi har prøvet så vidt muligt at lave hele databasen i 3 normalform, ud fra de attributter vi har valgt at tage med.



5. Når man godkender sit køb bliver det indkøbskurv objekt der ligger i session nulstillet.
6. Vores sikkerhed ligger i at vi tjekker om når man logger ind tjekker den databasen for username og password og tjekker om det stemmer overens med det du har skrevet. Og hvis man prøver at gå på en side via url så bliver man smidt ud på errorpage.jsp da man ikke har indtastet noget login, men ellers har vi ikke haft den store tid til at fejlhåndtere på de forskellige scenarier der kunne udfolde sig.
7. vi har været meget enige om at vores side ikke skulle have alt for mange under sidder, så vi har gjort alt hvad vi kunne for at holde brugeren så meget så muligt på index siden
8. vi har også fået lavet det sådan at der er ikke nogen inserts i databasen før du bekræfter at du vil købe de cupcakes og trykker confirm, da vi helst vil have så få så mulige database udtræk og inserts
9. vi har også sørget for at når man logger ind er vil man efter rolle kunne se forskellige ting, så f.eks. bliver du smidt ind på dashboard.jsp hvis du logger ind med rollen admin, og den kan du ikke tilgå hvis du er customer/kunde

Proces:

Fra start af havde vi en meget god plan om hvordan vi skulle starte, men faktisk ikke den store plan efterfølgende, vi startede mandagen med at få lavet mockups hvor vi tog udgangspunkt i den startkode vi havde fået afleveret og fik lavet nogle undersider etc. Vi fik også lavet et ER diagram ret hurtigt så vi hurtigt havde et rigtig godt udgangspunkt. Derefter planlagde vi at få helt styr på Git inden vi gik i gang, så det brugte vi en halv dag til at få styr på, og derefter kunne vi begge lave vores egne dele af projektet og efterfølgende føre det ind i det færdige projekt.

I praksis gik det faktisk rigtig godt, dog kunne vi godt have brugt lidt mindre tid på selve udseendet og mere tid på funktionalitet, så vi måske bare startede med en skrabt udgave, og derefter kunne kigge på det visuelle.

Vores gruppearbejde var alt i alt rigtig godt, men stadig nyt, så vi skulle lige finde ud af hvad vi begge skulle begive os ud i osv.

Casper: jeg har personligt lært at jeg oftest hurtigt nemt kan tage lidt for meget over, fordi jeg har en ide i mit hoved, som bare vil ud, det er noget jeg bestemt vil arbejde på, derudover er jeg rigtig hurtig til bare at kode løs, uden at tænke alt for meget over hvad der senere skal implementeres som så måske skal refactor senere.

Mads: Jeg har af tidligere erfaring lært at det kan være svært, når man arbejder på noget nyt i et projekt, at bedømme hvor svært og hvor lang tid ting tager. Derfor er det vigtigt hurtigt at starte og finde ud af hvilke ideer som virker og kan laves, og hvad som skal laves om, så man ikke bygger en masse som man først til sidst finder ud af, ikke virker eller kan lade sig gøre. I dette projekt har vi arbejdet ud fra dette, og gik hurtigt i gang med prototyper kort tid efter vi havde lavet vores første diagram over hvordan appen skulle se ud, så vi kunne lave ændringer hvis noget ikke passede ind. Det har gjort at vi ikke har haft nogle særlige problemer sent i projektet eller omskrivelser af ting som næsten var skrævet færdigt.