

CUPCAKE PROJEKT

Zack Ottesen – cph-zo27@cphbusiness.dk - Chunti
Louise Bengtsen – cph-lb352@cphbusiness.dk - DetBareLouise



1. APRIL 2022
Klasse B

Indholdsfortegnelse

Indledning	2
Baggrund.....	2
Teknologi	2
Krav.....	3
Aktivitetsdiagram.....	4
Domænemodel og EER diagram.....	5
Navigationsdiagram	6
Sikkerhed	6
Brugertest	7
Status på implementation	7
Proces	7

Indledning

Denne rapport omhandler et projekt, hvor der skulle laves et website til bestilling af cupcakes. Dette er udarbejdet ved, at vi har modtaget information omkring kundens behov og krav. Ud fra det har vi skulle tage nogle valg omkring sikkerhed, teknologi og opbygning. Selve rapporten er formidlet til en fagfælle.

Baggrund

Vi har landet en vigtig opgave fra Olsker Cupcakes. Det er endnu et dybdeøkologisk iværksættereventyr fra Bornholm, som har ramt den helt rigtige opskrift. Et par hipstere fra København har været forbi bageriet, og indsamlet nogle krav og lavet en halvfærdig mockup af en tænkt forside.

Som kunde skal man kunne oprette en profil på hjemmesiden. Med den profil skal man kunne bestille og betale cupcakes med valgfri bund og top. Ydermere skal man kunne tjekke sin ordrehistorik og fjerne ordrer. Som administrator skal man kunne indsætte penge på en kundes konto, se alle ordrer i systemet eller fjerne en ordre, hvis denne er ugyldig.

Teknologi

SDK 11.0.14

Maven 4.0.0

MySQL 8.0

Tomcat 9.0.60

openjdk-16 16.0.2

Der er blevet brugt Java til hoveddelen af projektet. Maven til at hjælpe for Java. Tomcat er blevet brugt som virtuel lokal server. MySQL er brugt som database og bruges også til at gemme generelle data, der skal bruges for den individuelle bruger.

Krav

De følgende krav fra kunden var henholdsvis:

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 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 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).

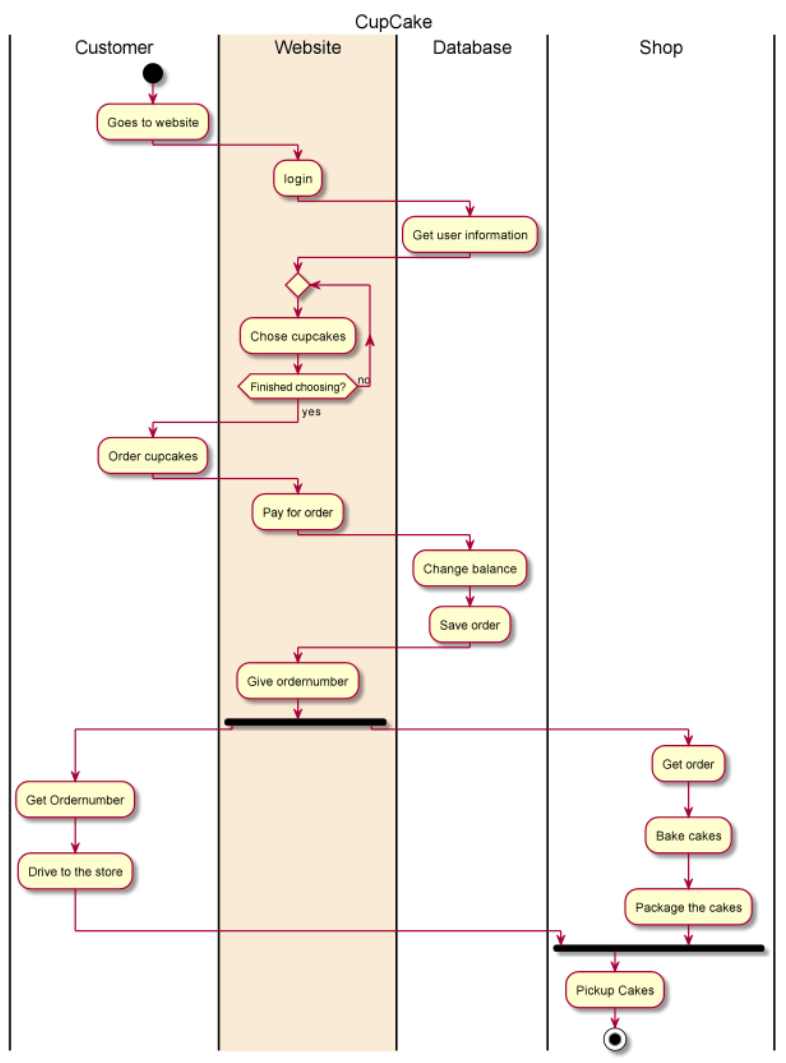
US-6: Som administrator kan jeg se alle ordrer i systemet, så jeg kan se hvad der er blevet bestilt.

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.

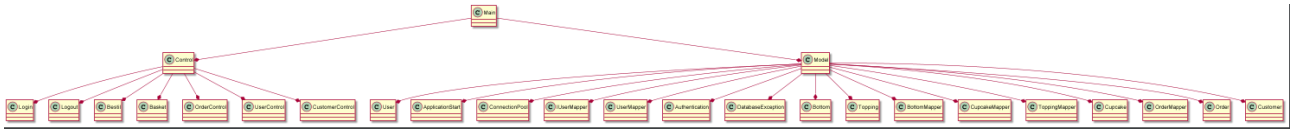
US-9: Som administrator kan jeg fjerne en ordre, så systemet ikke kommer til at indeholde ugyldige ordrer. F.eks. hvis kunden aldrig har betalt.

Aktivitetsdiagram

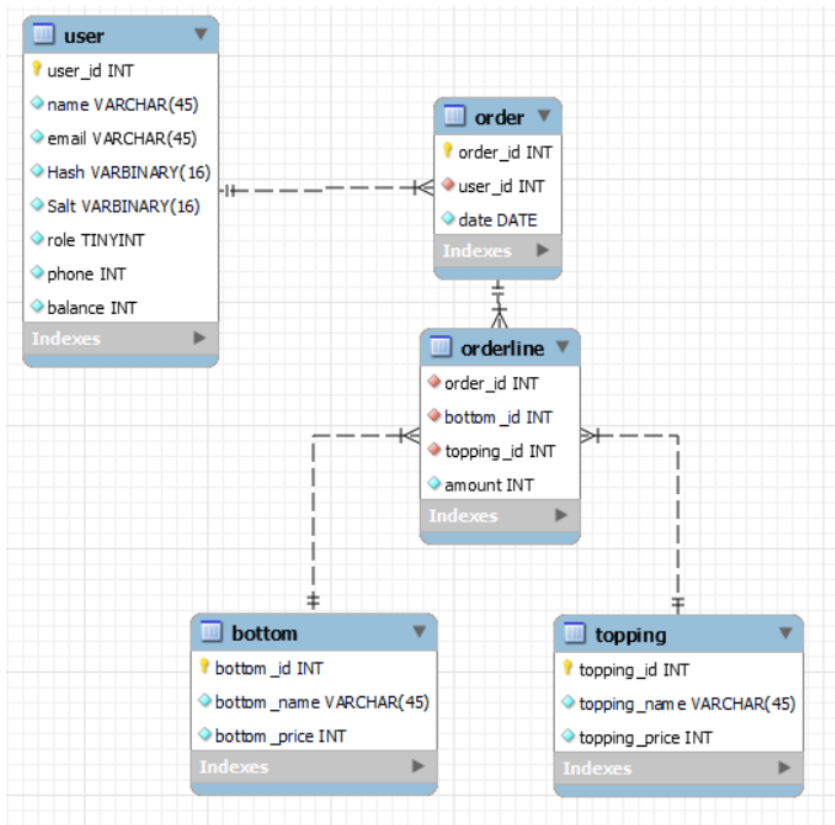
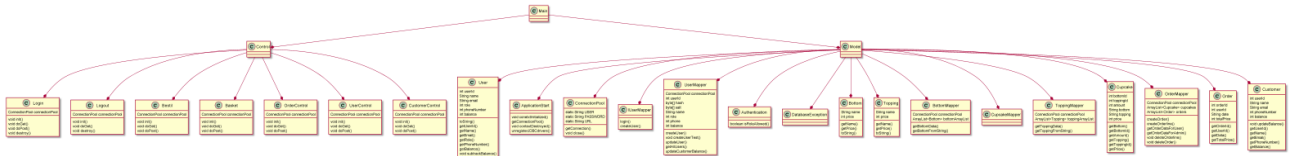


Aktivitetsdiagrammet starter fra, kunden går ind på hjemmesiden til, når de skal bestille varer. Når de er færdige med at vælge, så skal de betale for dem. Der går vi så ind i databasen, trækker pengene fra deres balance, gemmer ordren og giver dem et ordrenummer. Så fork'er den ud, at kunden skal køre ud til butikken, mens butikken får ordren, bager og pakker kagerne. Til sidst slutter fork'en i, at kagerne bliver hentet af kunden i butikken.

Domænemodel og EER diagram

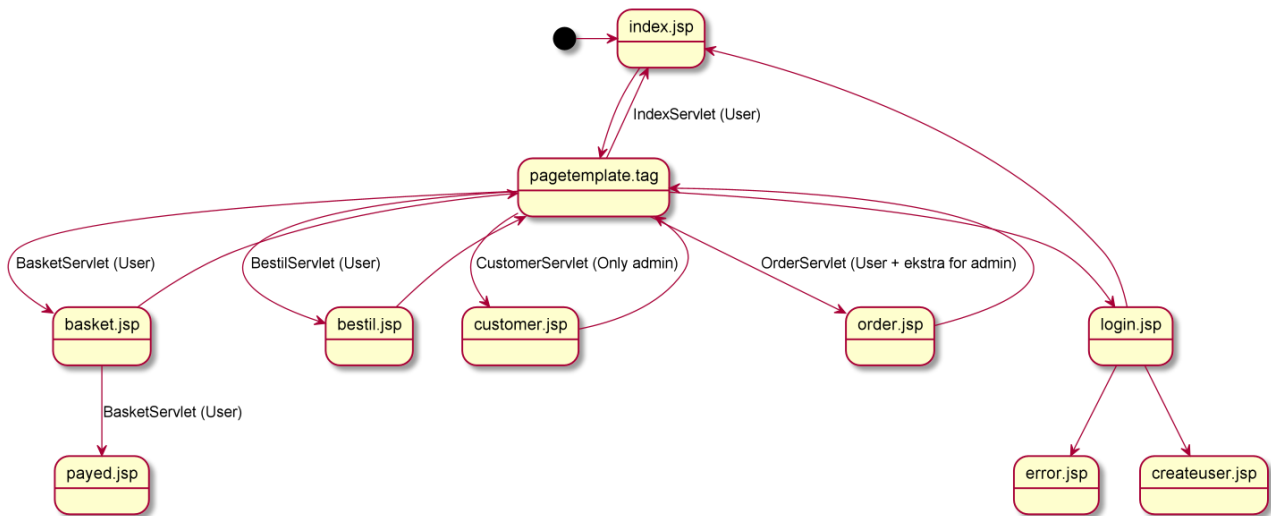


Her er en overordnet domænemodel over vores program. Der er ikke blevet lavet noget omkring nedrivning eller multiplicitet.



Orderline har ikke et automatisk genereret ID. Den bruges til alle de individuelle enheder bestående af toppings og bunde i en ordre. Tabellen fungerer ved brug af order id, hvor den tilføjer et bottom id, topping id og et antal. Det samme order id kan gå igen flere gange i tabellen alt efter, hvor mange forskellige cupcakes der er på samme ordre. Hver eneste orderline kan slettes inden, ordren er gået igennem, men gemmes fast efter betaling. Medmindre adminen sletter ordren.

Navigationsdiagram



Der er en fælles navigationsbar, som ændrer sig efter, hvilken bruger der er logget ind. Som kunde har vi mulighederne bestil, ordre og kurv. Udover det kan man se sin egen mail. Som administrator kan man se det samme som en kunde samt en ekstra bjælke, som hedder kunder. Ved at klikke på ordre som administrator, får man det fulde ordrekatalog fra databasen i stedet for den pågældende kundes ordrehistorik.

Sikkerhed

Lige fra starten valgte vi at tænke på kundernes sikkerhed ved at bruge hjemmesiden ift. opbevaring af kodeord. Vi ser det nemlig som et problem, at koden bliver gemt som en ren streng i databasen i tilfælde af et hackerangreb eller en korrupt administrator. Vi har derfor sikret kunderne ved at gemme deres kodeord som en krypteret hash bytearray med et vedlæggende bytearray som et salt. Det forgår ved, at vi tager det indtastede kodeord, lægger et tilfældigt genereret salt på for at forlænge kodeordet og til sidst bliver det hele kørt igennem en envejs krypteringsalgoritme. Dette gemmes i databasen sammen med det på gældende salt, der blev genereret ved til kunden. Siden det er en envejskryptering, kan kundens kodeord under login kun tjekkes ved at tage det indtastede kodeord, hente det gamle salt fra databasen, sende det hele igennem den samme envejskrypteringsalgoritme og til sidst tjekke, om den nye kryptering stemmer overens med den gamle kryptering, der ligger i databasen.

Brugertest

Vi lavede en brugertest for at se, hvor brugervenlig vores hjemmeside var. Ud fra brugertesten fik vi meget indsigt i, hvor vigtigt brugergrænseflade og generelt udseende af hjemmesiden kan hjælpe en bruger de rette steder hen. Vi kan meget hurtigt se inden for det første minut, at vi ikke har lavet den helt så brugervenlig, som vi havde forventet og skal hvert fald ikke mere ind i, hvordan vi visuelt fremstiller vores hjemmesider fremover.

<https://youtu.be/uki6p-8zWEM>

Status på implementation

Vi nåede at lave alle jsp siderne, som vi havde med i navigationsdiagrammet og alle CRUD metoderne til tabellerne. Vi stylede ikke vores sider, da vi havde problemer med CSS, så vi valgte bevidst kun at fokusere på backend. Vi fandt en "lille fejl", som dog er ubetydelig. Når man er logget ind som bruger og kigger på sin ordrehistorik, så står der "Welcome to the frontpage" som titel øverst.

Proces

Selve samarbejdet i ugen forløb som planlagt. Vi mødte klokken 9 hver morgen og arbejdede i ca. 9 timer om dagen på projektet. Vi satsede på at lave omkring 2-3 userstories om dagen og efter hver dag blev klassediagrammet opdateret.

Implementeringen af git var et langt og problematisk forløb. Det endte med, vi måtte bruge githubs egen userinterface i stedet for git. Vi prøvede at indsætte noget CSS i form af et baggrundsbillede på hjemmesiden. Dette fik hele programmet til at crashe. Selvom vi havde fjernet koden og ændret det tilbage, som det var før, så ville programmet ikke forstå, at det var fjernet. Dette gjorde, at vi måtte oprette et helt nyt projekt og starte forfra.

Ydermere blev der lavet en video af, hvordan hjemmesiden fungerer

<https://youtu.be/VOpSrqFTMH0>.