

The background of the slide is decorated with several stylized cupcake illustrations. In the top left, a yellow cupcake with white frosting and yellow flowers is partially visible. In the top right, a brown cupcake with white polka dots and white flowers is partially visible. In the middle right, a yellow cupcake with white frosting, colorful sprinkles, and a red heart is partially visible. In the bottom left, a blue cupcake with white frosting and white flowers is partially visible. In the bottom right, a pink cupcake with white frosting, white flowers, and a red heart is partially visible. The text is centered on the slide.

Datamatiker

Cupcake projekt

2. Semester

17. November 2022, kl. 12:00

Gruppe 3

Aleks, cph-ab279@cphbusiness.dk

Github: Aleks1075

Pelle, cph-pd94@cphbusiness.dk

Github: PelleErobreren

Indhold

Indledning	3
Baggrund	3
Teknologi	3
Krav	4
Aktivitetsdiagram	5
EER-Diagram	6
Domænemodel	7
Klassediagram	8
Navigationsdiagram	9
Særlige forhold	10
Status på implementation	10
Proces	10
Figma Mockup	11
Demonstration af brugergrænseflade	11
Link til github	11

Indledning

Formålet med dette webprojekt, Olskers Cupcakes, var at skabe en hjemmeside, der hentede information og gemte denne data i vores database i MySQL. Kravene til cupcake-opgaven er givet til os af vores undervisere Jon & Nikolaj.

Vi fik til opgave at lave en webshop, der sælger cupcakes fra Olsker Cupcakes. Kunden har bedt om en webshop til at hjælpe dem med at modtage og håndtere ordrer samt kunder.

Meningen med hjemmesiden, er at en bruger kan gå ind på hjemmesiden, oprette en brugerprofil

og derefter logge ind med den oprettede brugers brugernavn og password.

Baggrund

Virksomheden er et bageri, Olskers, der fremstiller cupcakes. Olskers Cupcakes ønsker en webshop, hvor kunderne kan bestille cupcakes og de kan se ordrerne i systemet.

Formålet med hjemmesiden er, at en bruger kan gå ind på hjemmesiden, oprette en brugerprofil og derefter logge ind med den oprettede brugers brugernavn og adgangskode.

Efter at brugeren har registreret sig på hjemmesiden, kan han lave sine egne cupcakes ved at vælge mellem forskellige bunde og toppe. Brugerens ordre sendes derefter til en database, så alle ordrer kan holdes styr på.

Som administrator skal det være muligt at se, redigere og slette ordrer samt tilføje penge til kunders saldi. Administratoren skal også kunne slette og redigere en kunde.

Teknologi

- JetBrains IntelliJ 2022.2.3
- MySQL Workbench 8.0.31
- HTML5 og CSS
- JSP
- Tomcat 9.0.69
- Java SE Development Kit 19.0.1
- Figma

- Safari og Google Chrome
- JSTL
- Bootstrap (5.0)

Krav

US-1: Som kunde kan jeg bestille og betale for cupcakes med valgfri bund og top, så jeg senere kan køre forbi butikken 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 deres 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 brugernavn og adgangskode. Når jeg er logget ind, skal jeg kunne se min e-mail 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 bestilt.

US-7: Som administrator kan jeg se alle kunder i systemet og deres ordrer, så 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.

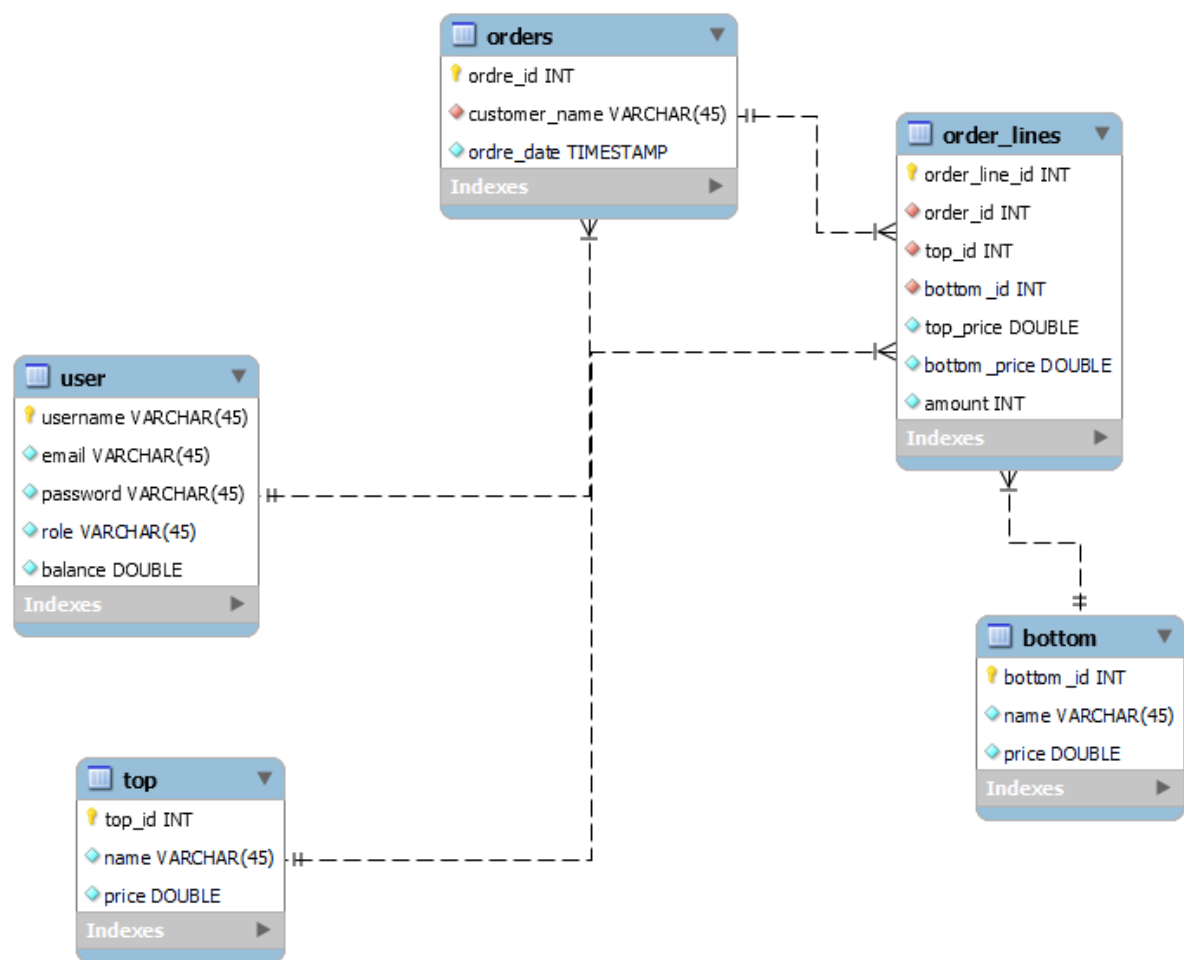
US-9: Som administrator kan jeg fjerne en ordre, så systemet ikke indeholder ugyldige ordrer. For eksempel hvis kunden aldrig har betalt.

Aktivitetsdiagram



Dette aktivitetsdiagram beskriver processen for en kunde, der logger ind på Olsker-webshopp, bestiller cupcakes og får det leveret af personalet i den fysiske butik. Webshoppens sørger for, at alle data i databasen er opdateret, samt sletter ordrer, der ikke er hentet.

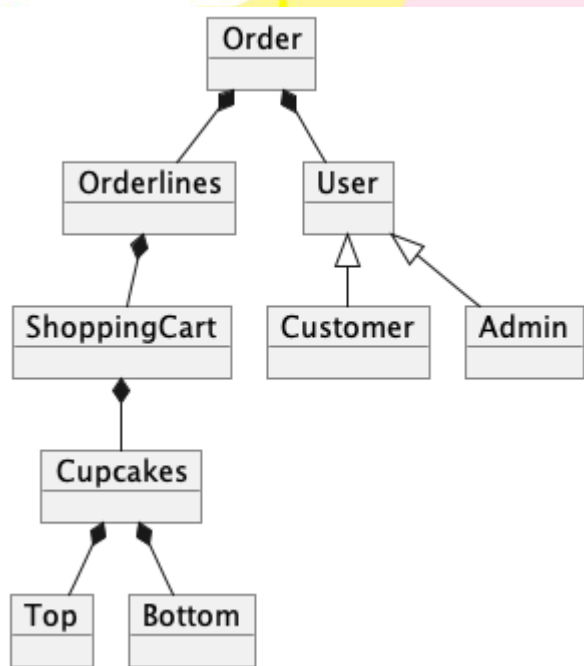
EER-Diagram



EER-diagrammet hjalp Aleks og jeg med at forstå grundlaget for de data/informationer, der skulle gemmes i vores database. Det var nemt at sætte op og også visuelt hjalp det, så vi begge kunne forstå det. EER-diagrammet ændrede sig dog undervejs.

I sidste ende var det vigtigt at kortlægge den grundlæggende struktur af vores database, før vi kodede noget, for at sikre, at jeg og Aleks var på samme side, så vi vidste præcist, hvor vores data skulle gemmes, samt sikre, at vi brugte de samme variabelnavne i IntelliJ (dette har givet os problemer før).

Domænemodel



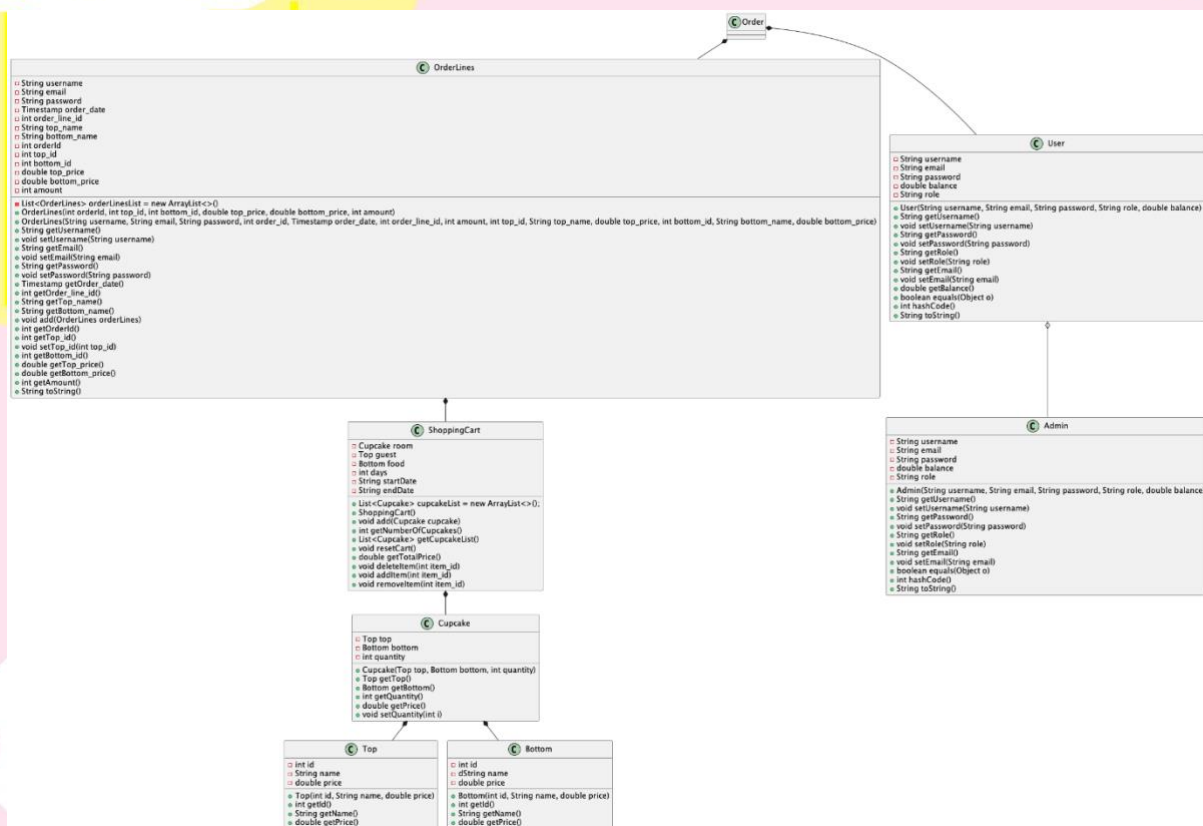
En af de første modeller, vi havde lavet, var domænemodellen, før vi overhovedet begyndte at kode. Domænemodellen hjalp os med at få et overblik over, hvad systemet skulle håndtere og herunder hvordan hierarkiet kommer til at se ud.

Det var vigtigt at repræsentere de forskellige enheder i systemet, og hvordan de var involveret i forhold til hinanden. Modellen brugtes primært til refleksion og for at få et bedre overblik over, hvordan programmet skulle køre.

Vores domænemodel viser, at du kan have to mulige brugere i programmet, nemlig "kunder" og "admins".

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

Klassediagram



Vores klassediagrammer illustrerer vores systems struktur på en detaljeret måde og viser dets egenskaber, operationer samt dets indbyrdes relationer.

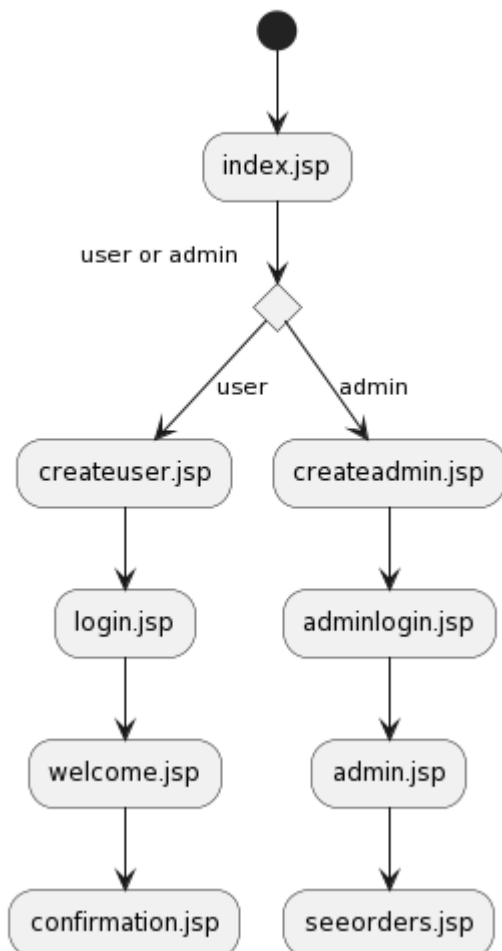
Under analysen fremstod de som domænemodellen, hvor vi forsøgte at skabe en repræsentation af virkeligheden.

Men under implementeringsfasen blev diagrammet brugt til at generere kildekode.

Klassediagrammet var en vital del af vores cupcake-projekt, og de har indtil videre dannet grundlaget for alle vores softwareprogrammer.

Navigationsdiagram

Navigationsdiagram Olskers Cupcakes



Vi brugte navigationsdiagrammet i designfasen til at give os et overblik over, hvordan hele systemet ville hænge sammen, og hvordan brugeren og administratoren forventes at navigere mellem forskellige grænseflader.

Navigationsdiagrammet var især nyttigt for os, fordi der er mere end én måde at navigere på vores hjemmeside.

Særlige forhold

Som kunde eller admin, gemmes man i sessionen og ligeså med e-mail for den bruger, der er logget ind. Imens når man skal hente query fra databasen såsom userlist eller itemlist gemmes i request.

Når man logger ind og opretter en bruger, valideres brugernavnet som username.

Adgangskode i log ind vises ikke, men er skjult under indtastning. Ved oprettelse af bruger skal adgangskoden bekræftes som den samme i begge indtastningsfelter.

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

Når man har afgivet ordren, stopper sessionen, men man kan stadig gå tilbage og se de besøgte sider, men man kan ikke lave input såsom at bestille flere cupcakes. Man vil kunne se, at du skal logge ind igen.

Status på implementation

Vi har generelt valgt at implementere funktion frem for design, da vi kom lidt bagud pga. US-1. Det betød, at vi havde mindre tid til at style websiden.


US-7 & US-9 er ikke implementeret, da vi simpelthen løb tør for tid, og de viste sig at være mere udfordrende end som så.

Proces

Den første dag satte Aleks og jeg os ned og etablerede startkoden, som Jon havde udarbejdet og fik sat projektet op, så vi kunne starte user-story 1. Vi blev enige om at samles hver dag for at diskutere opgaverne, der var blevet bøvlet med aftenen før, så vi begge havde set koden.

Derefter fortsatte vi med en daglig arbejdsgang, hvor user-stories blev distribueret.

Skoledagene udviklede sig sådan, at man gjorde status og derefter kodede videre med hver sin brugerhistorie men ved siden af hinanden, så man kunne spørge om hjælp i gruppen.



Arbejdsgangen var ret flydende, da vi ofte sprang frem og tilbage mellem de forskellige brugerhistorier. Dette fungerede fint for os og holdt os i gang, da vi følte, at vi af og til slog hovedet mod væggen, især med user-story 1.

Projektet gik i sin helhed rigtig fint, selvom der var nogle specifikke opgaver, der viste sig at være mere indviklede end først formodet, hvilket endte i mere arbejdstid end forudset.

For at undgå tilsvarende hovedpiner kunne opgaverne have været endevendt mere dybtgående, inden projektstart, og givet skøn over, hvor mange arbejdstimer de afkræver, for at de kunne klares.

Når vi ser tilbage, har vi kunnet se, at brugerhistorie #1 var betragteligt mere krævende end de andre.

Fremadrettet vil vi gerne bruge mere tid på at dele de store brugerhistorier op, så de nemmere kan fordeles mellem os som et 2-mandshold.

Figma Mockup

<https://www.figma.com/file/fTHA09IZHrHjuVMMJOqFa/Untitled?node-id=0%3A1&t=TeVYGrqrq4D3cFDE-1>

Demonstration af brugergrænseflade

Her er et link til demonstration af projektet. Der ageres først som kunde på siden og efterfølgende som administrator.

<https://youtu.be/1F6HxZ3hxe8>

Link til github

<https://github.com/Aleks1075/Cupcake-projekt>