

# Cupcake projektet

Deltagere:

- Anders Meinicke, [cph-am423@cphbusiness.dk](mailto:cph-am423@cphbusiness.dk), github navn: Anders Meinicke, klasse/gruppe A
- Rasmus Taul, [cph-rt91@cphbusiness.dk](mailto:cph-rt91@cphbusiness.dk), github navn: rasm445f, klasse/gruppe A

Dato: Startede 4 april 2022, endte 21 april 2022.

## Table of Contents

<b>Indledning.....</b>	<b>3</b>
<b>Baggrund.....</b>	<b>3</b>
<b>Teknologivalg .....</b>	<b>3</b>
<b>Krav .....</b>	<b>4</b>
<b>Aktivitetsdiagram .....</b>	<b>5</b>
<b>Domæne Diagram.....</b>	<b>6</b>
<b>EER og UML-diagrams.....</b>	<b>7</b>
<b>Særlig forhold .....</b>	<b>8</b>
<b>Status på implementering .....</b>	<b>9</b>
<b>Proces .....</b>	<b>9</b>
<b>Demo video.....</b>	<b>10</b>

## Indledning

Vores opgave handler om at skabe en hjemmeside til det bornholmske bageri Olsker Cupcakes. I projektet skulle der bruges html, servlets og jsp og der skulle implementeres en database. Vi skulle bruge disse værktøjer til at lave et program, hvor man kunne logge ind som en bruger og bestille en vare eller som en admin der kunne se alle de forskellige bestillinger der var lavet.

## Baggrund

Vi har fået til opgave at fremstille et produkt i form af en hjemmeside/system til det lille nystartede bornholmske bageri Olsker Cupcakes, der ønsker at sælge deres cupcakes via internettet. Målet for bageriet er, at kunder skal kunne bestille varer og se den samlede totalsum af ordren på hjemmesiden. Derefter er det muligt for kunden at betale og hente ordren i bageriet. Ud over det er der også andre funktionelle krav til produktet. Olsker Cupcakes ønsker at kunder skal kunne oprette sig som bruger i deres system med en e-mail og et kodeord. Der skal være en funktionel indkøbskurv, hvor kunden kan se hvad der er bestilt og hvor mange penge man som kunde skal betale. Olsker Cupcakes skal kunne oprette sig selv i systemet med det formål at holde øje med samt administrere ordrer.

## Teknologivalg

- **Intellij:** Dette program er meget nyttigt til at skabe et let overskueligt projekt,. Det har en masse brugervenlige funktioner, der gør det lettere at skabe nye sider til ens projekt, men også at finde fejl i projektet.
- **Java:** Vi anvender Java som vores primære kodesprog i denne opgave. Den primære årsag er, at medlemmer i gruppen har størst kendskab til Java. Vi endte også med at få adgang til en del teknologier som er afhængige af Java, hvilket fik andre kodesprog til at være mindre ideelle.
- **Mysql:** Mysql er et meget nyttigt program til at skabe databaser, som vi let kan tilslutte til vores projekt,. Det indeholder også andre funktioner, fx at kunne skabe modeller for ens database, en af disse slags modeller er et EER diagram.
- **Sql:** Dette er sproget vi anvender til at skabe og opdatere vores databaser. Vi har ikke lært om andre sprog der kan gøre det, så vi bruger sql i dette projekt.
- **Tomcat:** Vi bruger tomcat til at opsætte test servere, så vi hurtigt kan se hvordan vores projekt virker når det bliver kørt. Disse visuelle elementer hjælper med at få et overblik over evt. mangler. Det hjælper også med at finde fejl som ikke er direkte inde i koden.
- **JDBC:** JDBC bliver anvendt til at tillade hurtig adgang til vores database. Vi kan sende String værdier direkte til vores database, fx til at hente eller opdatere vores informationer i databasen.

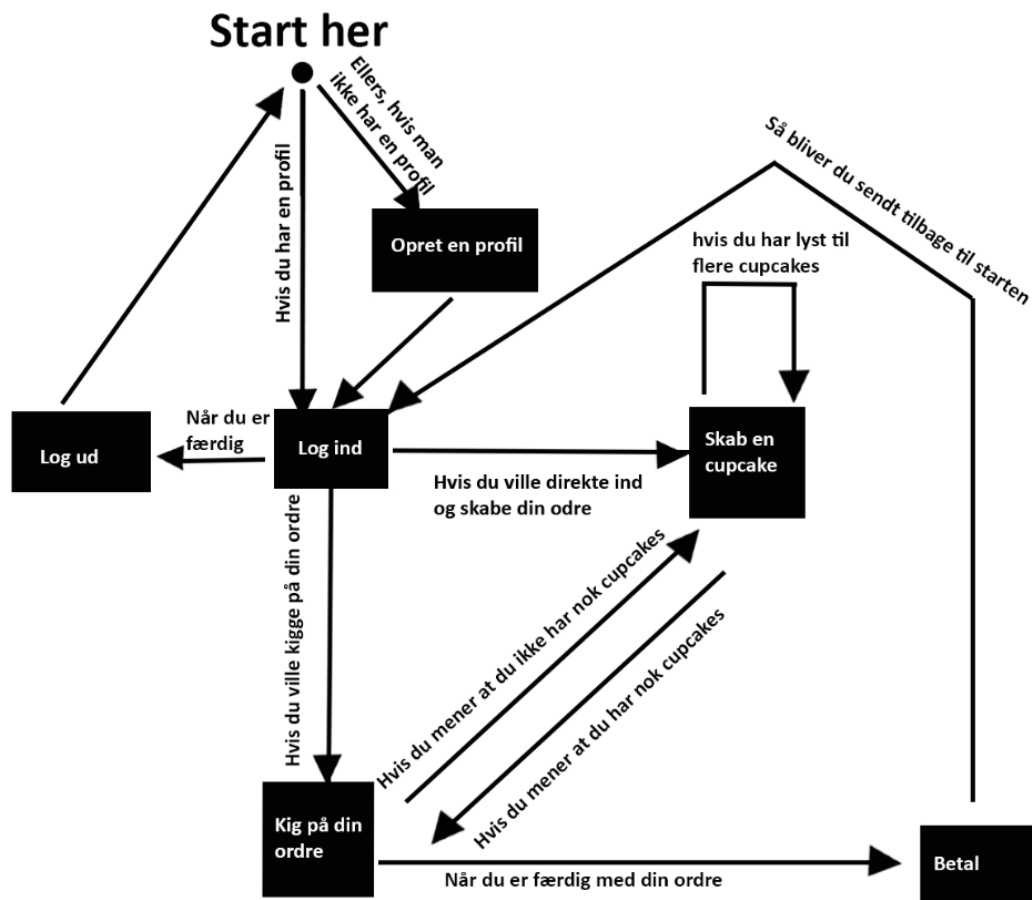
- **jsp:** Dette bruger vi som den visuelle del af vores projekt. Med jsp kan vi sende informationer rundt i projektet og bruge dem til at skabe noget en bruger lettere vil kunne forstå, fx et spørgeskema eller en liste af objekter.
- **Servlets:** Servlets er specifikke klasser vi anvender til at indtage information og udføre funktioner efter hvad en bruger vil gøre, såsom at bruge informationer en bruger indsatte, til at se om de har en profil i vores database.
- **CSS:** Dette kodesprog bliver anvendt til at gøre vores hjemmeside mere behagelig at se på.

## Krav

Vi har fået disse krav til projektet af Olsker Cupcakes:

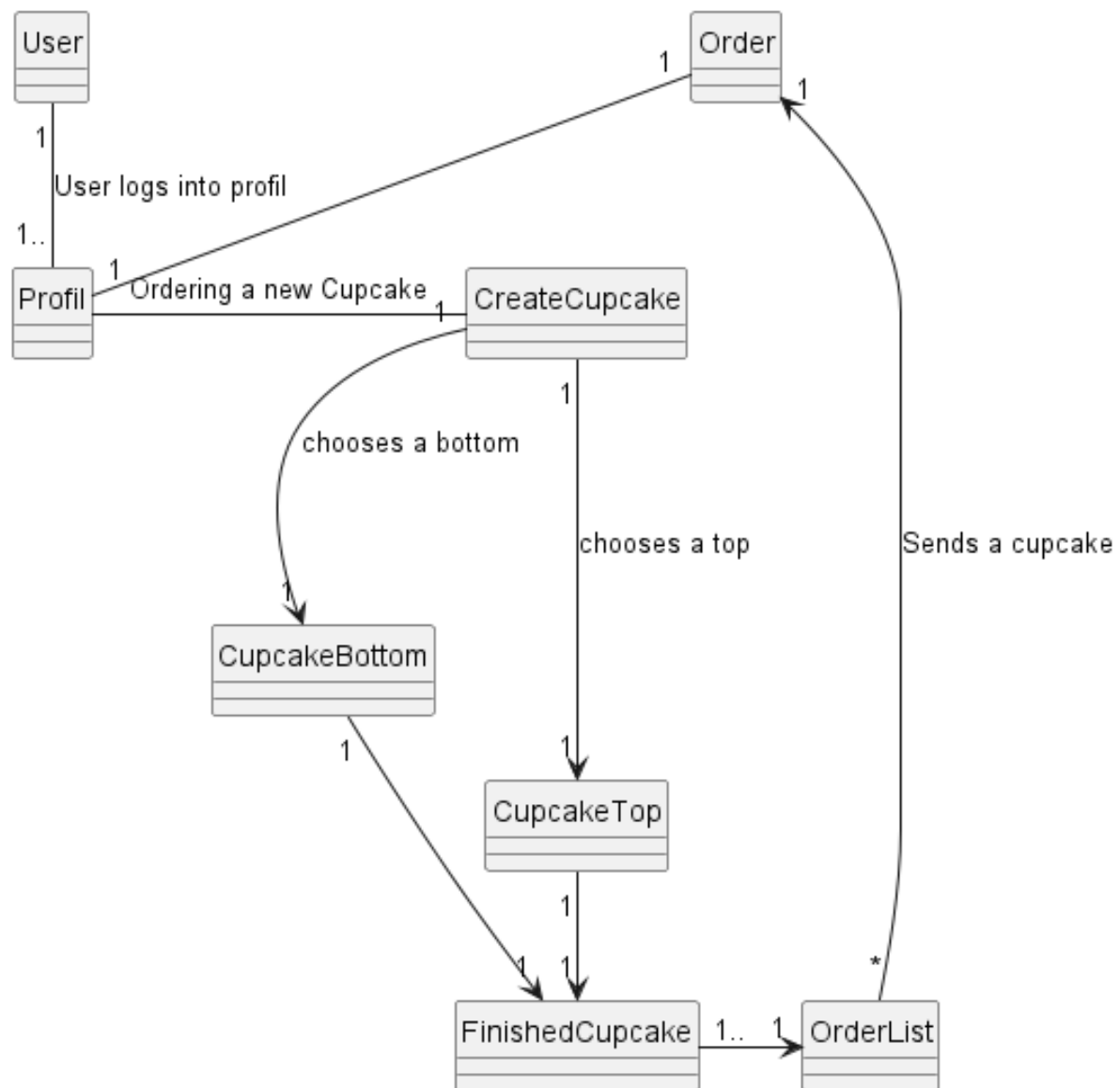
- ~~Som kunde kan jeg bestille og betale cupcakes med en valgfri bund og top, sådan at jeg senere kan køre forbi butikken i Olsker og hente min ordre.~~
- ~~Som kunde kan jeg oprette en konto/profil for at kunne betale og gemme en en ordre.~~
- ~~Som administrator kan jeg indsætte beløb på en kundes konto direkte i MySQL, så en kunde kan betale for sine ordrer.~~
- Som kunde kan jeg se mine valgte ordrelinier i en indkøbskurv, så jeg kan se den samlede pris.
- 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).
- ~~Som administrator kan jeg se alle ordrer i systemet, så jeg kan se hvad der er blevet bestilt.~~
- 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.
- Som kunde kan jeg fjerne en ordre fra min indkøbskurv, så jeg kan justere min ordre.
- Som administrator kan jeg fjerne en ordre, så systemet ikke kommer til at indeholde udgyldige ordrer. F.eks. hvis kunden aldrig har betalt.

## Aktivitetsdiagram



Dette var den primære ide til, hvordan en kunde skulle navigere på hjemmesiden. Diagrammet inkluderer kun det nødvendige, da vi ikke kunne komme op med andre ting som hjemmesiden skulle kunne gøre for kunden.

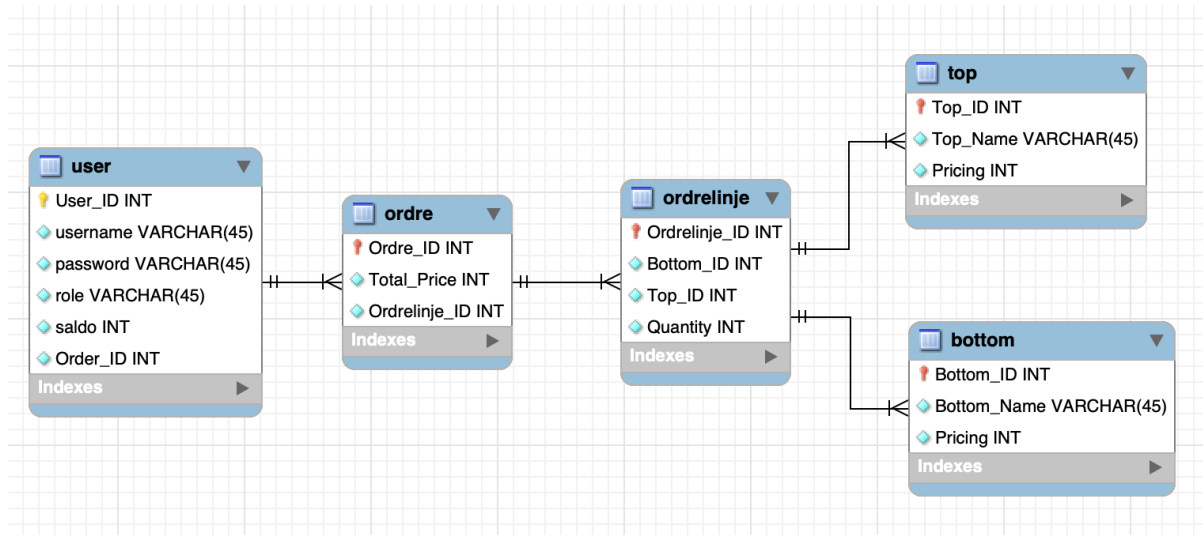
## Domæne Diagram



Vores design til domæne diagrammet var meget simpelt. Vi havde en bruger der skulle logge ind på en profil, den profil skulle så kunne se sin ordre eller tilføje endnu en vare til ordren. Bruger skulle kunne vælge en af både toppings'ne og kagebundene, det ville give en færdig vare, som man giver en ordreliste. Antallet af færdige cupcakes man sender ind i ordrelisten kan være en eller flere, da man ikke har lyst til konstant at skabe den samme cupcake.

Ordrelisten sendes til profilens ordre, så kunden så kan se alt hvad de har bestilt.

## EER og UML-diagrams



Undervejs i database processen fokuserede vi på at skabe et simpelt system til at håndtere data.

Vores PK ligger i user og hedder user\_ID. På hjemmesiden starter man med at logge ind som bruger. Derefter kan man sammensætte sin ordre med en Top, en Bottom og et antal.

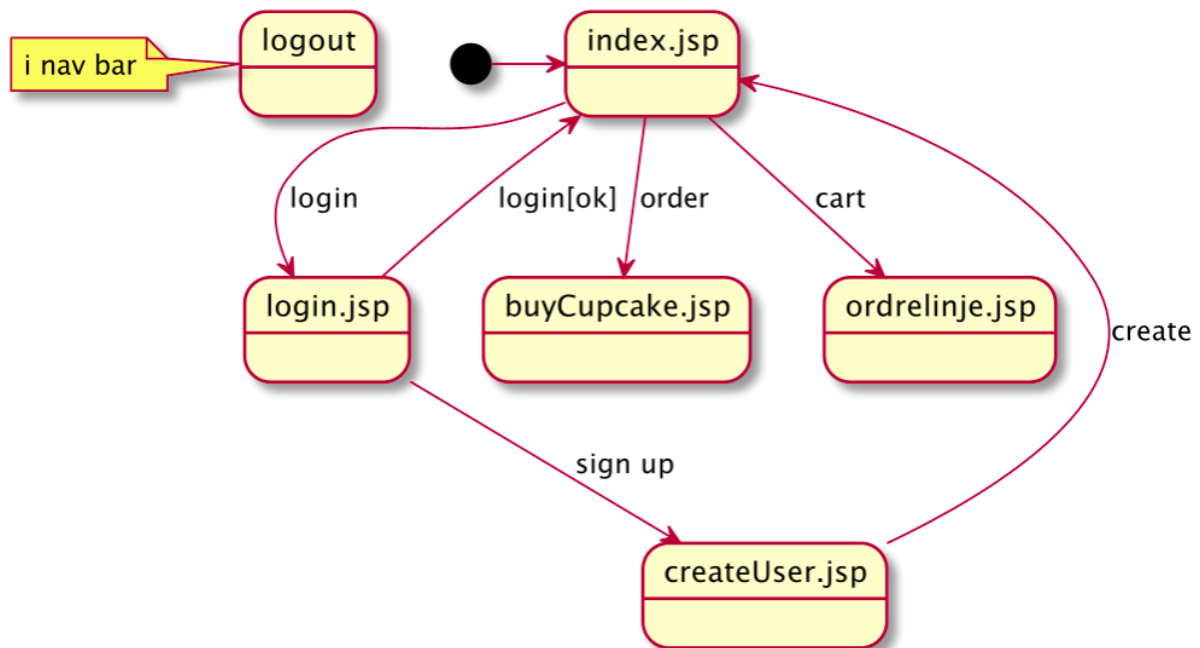
Efter kunden har trykket 'Tilføj til kurv' vil det blive til en ordrelinje hvori ID'et fra Top og Bottom bliver til en fremmednøgle. Antallet er også gemt i ordrelinje med navnet Quantity.

Forholdet mellem Top og Bottom er en 1:1 relation så en ordrelinje kun kan indeholde 1 top og 1 bottom.

Hvis kunden er færdig med at handle bliver det til en samlet ordre med en Total\_Price, Ordre\_ID og en fremmednøgle ordrelinje\_ID. Ordre\_ID kommer derefter i user som en fremmednøgle. Dvs at man med User\_ID kan finde den ordre som den kunde har bestilt.

Mellem ordre og user er der en 1:1 relation da 1 user kun kan have en ordre dog ville det være smart hvis user kunne have flere ordrer, i det tilfælde at kunden valgte at komme igen.

## Cupcake : state diagram



På hjemmesiden benytter vi en fælles navigations bar, hvori man kan navigere mellem ordre, login og cart. Hvis man er logget ind bliver login erstattet af logout.

Vi navigerer mellem jsp siderne med servlets. Vores servlets har samme navn som jsp siderne dog har vi erstattet “.jsp” med “Servlet” . Fx har vi navngivet den servlet der navigerer os til `buyCupcake.jsp` siden for `buyCupcakeServlet`. Dette har vi gjort for at lettere at finde rundt i programmet.

### Særlig forhold

Vi gemmer det meste af bruger dataen på session, vi gemmer deres brugernavn, deres ordre lister, og deres ordre, på den måde kan vi let printe ud information til brugeren som de gerne ville se, fx hvad de har bestilt. Anden information fx en ny brugers kodeord bliver gemt på vores database.

Der er to metoder vi bruger til at handle exceptions, den første bliver brugt, når der sker en fejl med at loade en side, der ville den så føre en til en ny side og forklare hvilken fejl skete, den anden metode bliver brugt når der sker med en funktion inde i siden, når det sker kommer der en besked op, på siden som siger at en fejl er sket, denne besked bruger mindre teknisk sprog, da den er mere ment til kunder der har glemt at gøre noget.

Vores validering af brugerens input, kommer i form af at der altid sker noget på siden. Vi opdaterer hele tiden sessionen når brugeren sender nogle informationer og der er ikke et tidspunkt hvor noget går igennem, uden også at ændre noget på hjemmesiden. Dog er nogle af disse ændringer mindre end andre.



Vi har ikke den største form for sikkerhed, det er kun det mest basale vi har der, Adgangskoden er blevet sat ind som en private konstant i vores bruger, med ingen direkte måde at ændre den på, så man er nødt til at hacke koden eller komme ind på vores database før at man få fat på dem.

Vi har lavet en masse ligne entities, som alle sammen høre sammen, med noget der eksistere i vores database, hvilket gør så vi ikke får nogle konflikter, når vi skal gøre noget mellem serveren og databasen.

## Status på implementering

Som produktet er lige nu, er det muligt at logge ind med en e-mail og et selvvalgt kodeord. Hver user har en rolle enten admin eller user. Man kan som kunde eller admin gå ind på cart og se alle ordrer i systemet. Det er muligt at sammensætte en cupcake med hvilken som helst top eller bottom og få prisen på dem i ordren. Det er også muligt at tilføje et antal af den sammensatte cupcake.

En fejl der blev fundet sent i programmets udvikling er, at efter man har foretaget en række handlinger vil hjemmesiden fryse (loader hele tiden). Vi konstaterede at vi ikke havde lukket nogle af "connections'ne" korrekt, hvilket betød at alle trådene programmet havde adgang til, blev brugt på samme tid, så der var ingen ledige til at hjælpe brugeren.

Vi løste dette problem ved at gå ind i den kode vi planlagde at gøre til mappers, hvor vi ændrede på metoden der lukkede dens connection efter den var brugt.

Der er desværre nogle funktioner som vi ikke er kommet i mål med i vores projekt. Der er ikke forskel på om du som user vælger en admin rolle eller en kunde rolle, selvom der er mulighed for det. Grundet denne manglende funktion kan alle der har en profil gå ind og se hvilke ordre der er i systemt via ordrelinje.jsp siden. Det er ikke muligt som kunde kun at se de ordre man har bestilt, men altså kun alle ordre i systemet. Som admin er det ikke muligt at se alle kunder i systemet og alle deres ordre. I navigationsbaren bliver der ikke vist den e-mail på useren der er brugt til at logge ind med.

## Proces

Vores arbejdsproces i dette forløb har udviklet sig således, at vi har mødt op på skolen hver dag så vi har kunne få hjælp af lærer samt medstuderende. At vi har mødt på skolen og har kunne få hjælp har gjort at vi ikke som sådan har siddet fast på noget tidspunkt. Hvorimod hvis vi sad hjemme/online ville vi skulle have brugt meget mere tid på at troubleshoot og samle information. Vi har så vidt muligt fulgt vores domænemodel for at skabe struktur og for at skabe et overblik over arbejdsprocessen.

I retrospekt skulle vi have været bedre til at udnytte tiden når vi sad fast. Vi havde nok fået gavn af at lave en to-do liste samt en plan over hvad der skulle være lavet til hver dag. Vi har undervurderet tiden vi havde til rådighed til at færdiggøre projektet og derfor har vi flere mangler end vi er tilfredse med.

### Demo video

**Link til en hurtig fremvisning af programmet:**

[https://www.youtube.com/watch?v=Py69M-IETYE&ab\\_channel=AndersMeinicke](https://www.youtube.com/watch?v=Py69M-IETYE&ab_channel=AndersMeinicke)