

Cupcake projekt 04/21



Lavet af:

● **Benjamin Ritthidech Sommervoll**

○ cph-bs202@cphbusiness.dk

○ <https://github.com/pl0xi>

○ Klasse: A

● **Daniel Anders Drobek**

○ cph-dd77@cphbusiness.dk

○ <https://github.com/DanielD99>

○ Klasse: A

● **Emil Ruby Andersen**

○ cph-ea120@cphbusiness.dk

○ <https://github.com/syntax001>

○ Klasse: A

● **Firat Kayaalp**

○ cph-fk99@cphbusiness.dk

○ <https://github.com/firatk00>

○ Klasse: A

● **Florian Michael Schaffer**

○ cphfs101@cphbusiness.dk

○ <https://github.com/Florian-Schaffer>

○ Klasse: A

Indledning	3
Baggrund	3
Kundens krav	3
Teknologivalg	4
Krav	5
Firmaets Håb	5
User Stories	5
Aktivitetsdiagram	6
Use Case model baseret på vores user stories:	7
Domæne model og EER diagram	8
Domænemodel:	8
EER Diagram:	9
Navigationsdiagram	10
Særlige forhold	11
Data gemt i session	11
Status på implementation	11
Proces	11

Indledning

I dette projekt arbejdes der med en webshop til en cupcake-forretning. Målet er at få et sammenspil mellem diverse programmer til at fungere således at kunderne via hjemmesiden kan placere ordre, samt se deres aktivitetshistorik ved hjælp af en database og *user-credentials*.

Desuden skal der også implementeres måder for en admin at se alle ordre og ændre status i tilfælde af nødvendighed. Projektets funktionalitet er beskrevet ved hjælp af diverse vedlagte diagrammer, og kan bruges til at færdiggøre projektets struktur og indhold.

Primær fokus i denne opgave er arbejdet med en online side og overførelsen af java kode samt sql databaser til en interaktiv side.

Baggrund

Cupcake projektet er specifikt programmeret til *Olsker Cupcakes*, med mulighed for fleksibilitet angående data (Toppings, bottoms) via. databasen.

Olsker Cupcakes er et bageri fra Bornholm.

Kundens krav

- Kunder skal kunne bestille cupcakes med valgfrie toppings/bottoms, samt betale og opsætte afhentning.
- Kunder skal kunne oprette profiler, så de kan betale og gemme deres ordre.
- Administrators skal kunne ændre på saldoen af kundernes profiler.
- Kunder skal kunne se deres valgte ordrelinjer i en indkøbskurv, hvori der skal være en "samlet pris" for de valgte ordre.
- Kunder og Administratorer der er logget på, skal på hver side have fremvist deres email.
- Administratorer skal kunne se alle oprettede ordre.
- Administratorer skal kunne se alle kunder oprettet i systemet, samt deres ordre.
- Kunder skal kunne fjerne ordre fra deres indkøbsliste.
- Administratorerne skal kunne fjerne ordre.

Teknologivalg

- IntelliJ IDEA Ultimate (2020.3.1) og IntelliJ IDEA Ultimate (2021.1)
 - IDE'et som er blevet brugt til at redigere/oprette og se filer:
 - JSP sider
 - Java classes
 - css filer
 - sql filer (Ændres også i MYSQL Workbench)
 - database data
- Adobe XD
 - Der er blevet brugt Adobe XD til at lave vores mockups.
 - Der er blevet brugt Bootstrap ved oprettelse af mockups.
- MySQL Workbench (8.0.22)
 - MySQL Workbench er blevet brugt til:
 - EER Diagrams
 - Database opsætning (Mest via. forward engineering af EER Diagram)
 - Se database data
- Code With Me (IntelliJ plugin) (203.6682.168)
 - Der er blevet brugt Code With Me til at samarbejde under projektet
- MySQL Community Server - GPL (8.0.22)
 - Database server
- VM/Droplet (Digital Ocean): Ubuntu, 20.04 (LTS) x64, 1vCPU og 25GB disk kapacitet
 - VM der er blevet brugt til at teste tomcat serveren.
- Apache Tomcat Server (9.0.44)
 - Java servlet
- JSTL (1.2)
 - Tag bibliotek, for at lave task i JSP sider
- Bootstrap (4.3.1) - CSS Bibliotek
 - Bootstrap er blevet brugt til at designe hjemmesiden
- JQuery (3.0.0)
 - JavaScript bibliotek
- Java (15.0.1)
 - Primær programmeringssprog for backend af projektet.

Krav

Firmaets Håb

Det endelige produkt skal for firmaet være en nem online tilgang for deres kunder til cupcake-forretningen. Dermed sikrer de sig at der er mulighed for udvidelse grundet den nemme tilgængelighed af deres produkter, og samtidig en oversigt over deres online-indtjening baseret på databaser.

User Stories

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 e-mail og kodeord. Når jeg er logget på, skal jeg kunne min email på hver side (evt. i topmenuen, som vist på mock up'en).

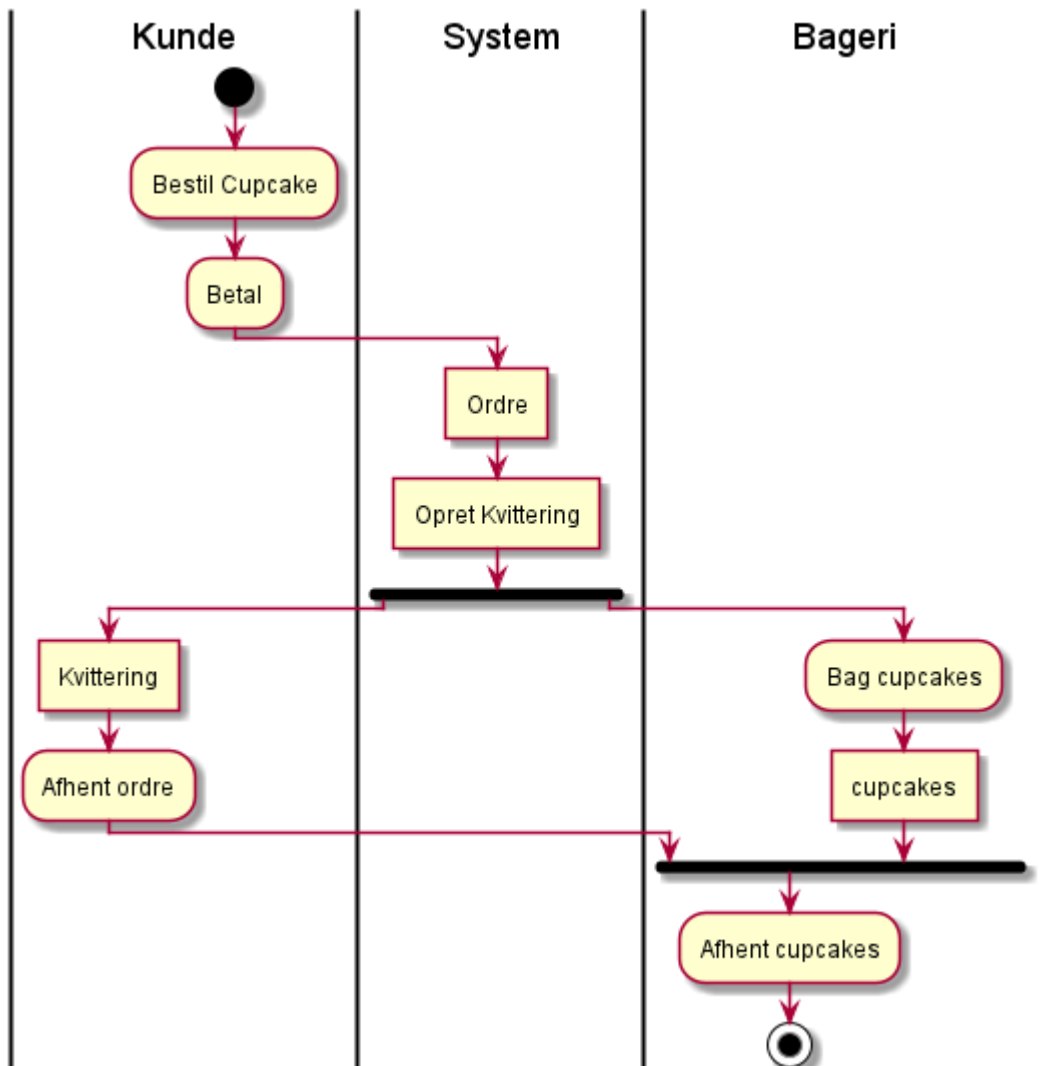
US-6: Som administrator kan jeg se alle ordrer i systemet, så jeg kan se hvad der er blevet bestilt.

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 består af 3 aktører, kunden, systemet (Hjemmesiden/Servlet) og bageriet.

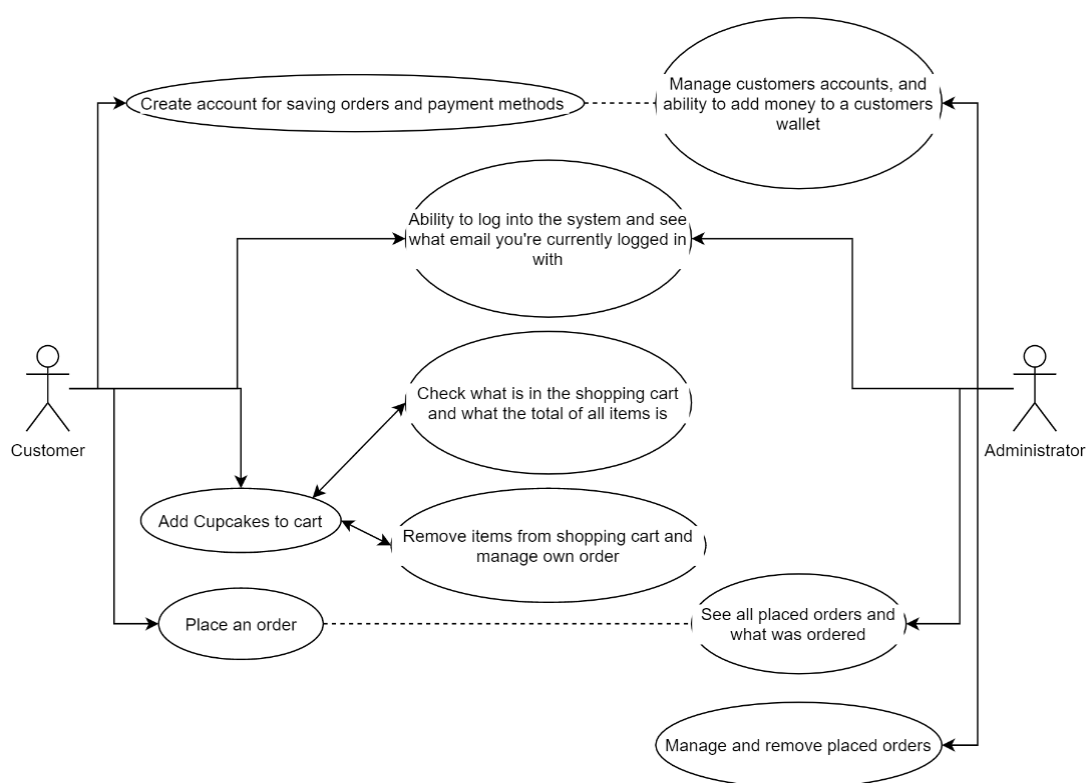
Det starter med at kunden via. systemet (Hjemmesiden) skal lave en ordre. Den nuværende betalings implementation er at kunden skal have penge på deres konto i systemet, ellers vil ordren ikke gå igennem.

Når kunden har lavet sin ordre, vil den blive gemt i systemet, hvori en admin skal kunne tjekke, via. en ordreliste, alle ordre. Systemet vil oprette en kvittering for kunden, hvorefter kunden kan afhente sin ordre i bageriet, ifølge med at bageriet laver ordren.

Use Case model baseret på vores user stories:

Use Case diagrammet er en visuel repræsentation af kravene der er pålagt gennem vores User Stories. Diagrammet giver en visuel præsentation af hvilke aktører vi har med at gøre og hvilke funktioner de skal kunne udføre i vores færdigarbejdede system. Use Case diagrammet giver os et godt overblik over, hvad systemet overordnet skal kunne.

Use Case Diagram

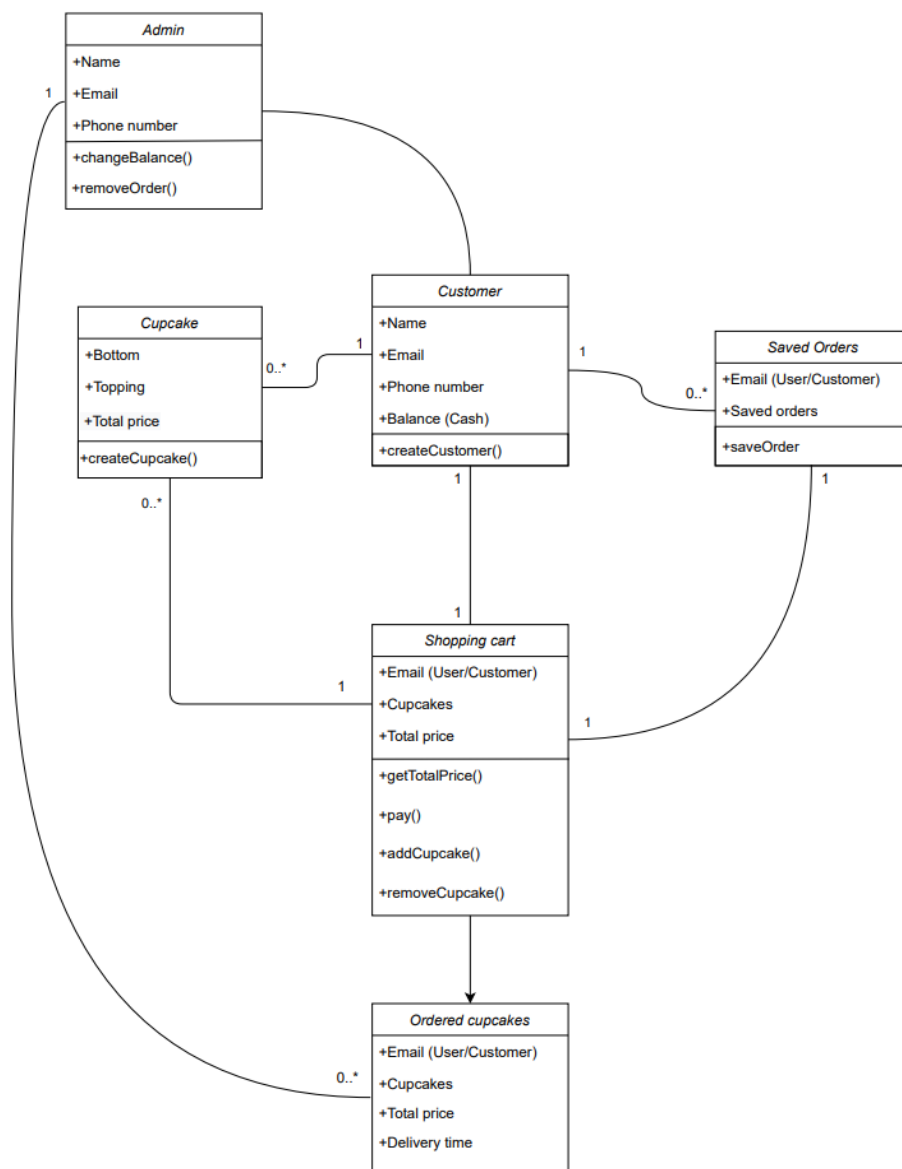


Domæne model og EER diagram

Domænenemodel:

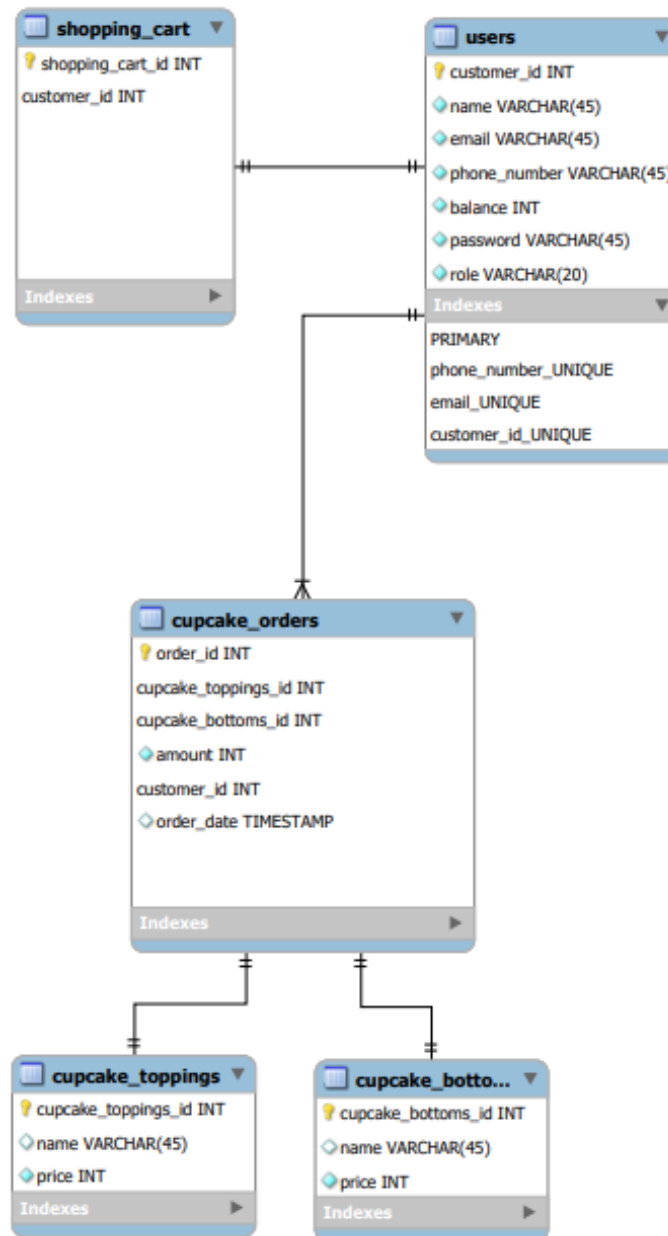
Diagrammet viser relationerne mellem de forskellige klasser hvor især kunde-, og indkøbskurv-klassen er i fokus da det er dem der bruges mest på websiden. I deres sammenhæng er der også en 1 til 1 relation som hovedsageligt sikrer sig at ordrene ikke kan blive tilgået af en anden kunde end den som har lavet ordren.

Der skal gøres opmærksom på at domænenemodellen er lavet i starten af projektet og ikke afspejler det færdige produkt.

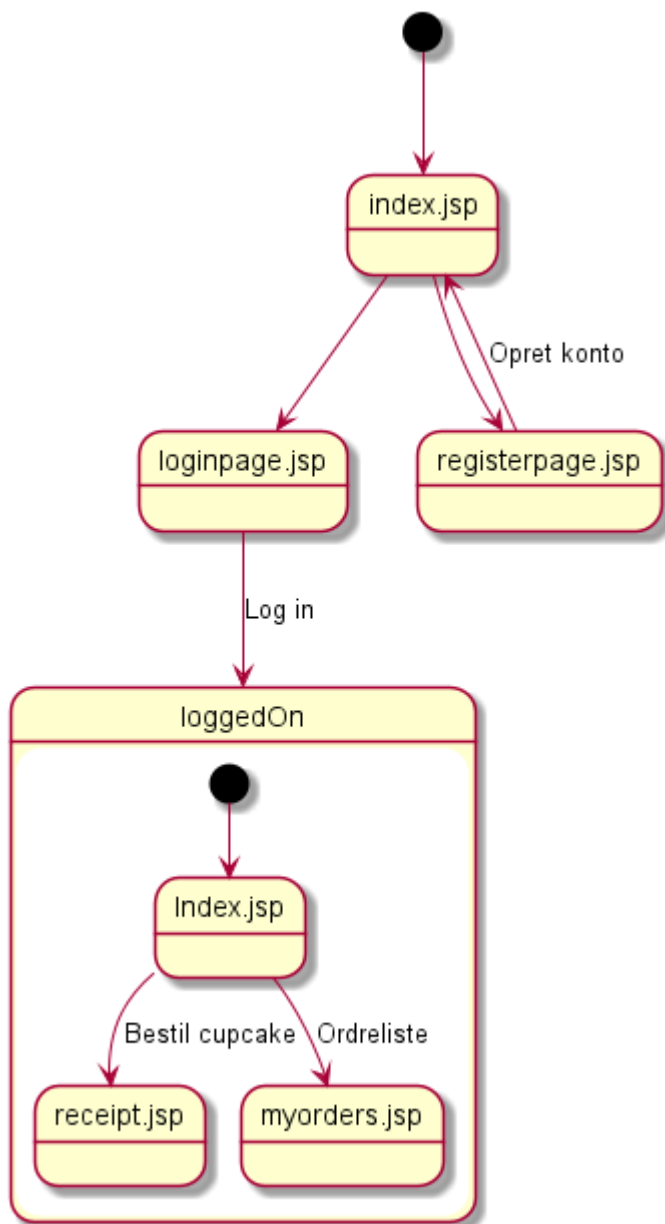


EER Diagram:

EER diagrammet er lavet i MySQL Workbench, hvorefter vi ud fra diagrammet har lavet *forward engineering* til at overføre diagrammet til databasen.



Navigationsdiagram



Hjemmesiden benytter sig af en fælles navigationsbar, som inkludere disse sider:

- Kræver ingen login:
 - Home (Index.jsp)
 - Sign up (registerpage.jsp)
 - Log in (loginpage.jsp)
- Kræver login:
 - Orders (myorders.jsp)

Særlige forhold

Data gemt i session

- User objekt
- role (Employee eller customer)
- email

Status på implementation

1. Vi har ikke nået at lave de sidste JSP sider i **US-3**
2. **US-4, US-5, US-6, US-7, US-8, US-9** er ikke blevet lavet
3. Ingen sider er blevet stylet
4. Hjemmesiden er ikke blevet hosted på en droplet

Proces

Planen for dette projekt var individuelt arbejde med hjælp for fælles gennemgåelse i gruppen af eventuelle problemer. Delopgaverne inden for projektets rammer blev uddelegeret via trello.

I praksis gik det meget efter planen med fælles fremmøde på discord, og mulighed for at spørge ind til hinandens kode og få hjælp. Grundet personligt valg af mødetidspunkt var vores arbejdstimer mere flexible som både var godt og dårligt, da der ikke var et særligt stort pres til punktlighed, men på den anden side kan trække lidt ind over de resterende planer uden for skoletiden.