



# Cupcake

D: 28 / 04 / 21

GitHub: <https://github.com/Mfknuksen/Cupcake>

Af:

Mads Teddy Francois Knudsen:

[cph-mk682@cphbusiness.dk](mailto:cph-mk682@cphbusiness.dk)

Github: Mfknuksen

Christoffer Munk Quakernaat

[cph-cq2@cphbusiness.dk](mailto:cph-cq2@cphbusiness.dk)

Github: ChristofferQ

Albert Emil Hyrde Jensen:

[cph-aj419@cphbusiness.dk](mailto:cph-aj419@cphbusiness.dk)

Github: Zero-slayer

Hold B gruppe 4

<b>Indledning:</b>	<b>3</b>
<b>Baggrund:</b>	<b>4</b>
<b>Teknologivalg:</b>	<b>5</b>
<b>Krav:</b>	<b>6</b>
<b>Aktivitetsdiagram:</b>	<b>7</b>
<b>EER diagram:</b>	<b>8</b>
<b>Domæne model:</b>	<b>9</b>
<b>Mockup:</b>	<b>10</b>
<b>Navigationsdiagram:</b>	<b>11</b>
<b>Særlige forhold:</b>	<b>12</b>
<b>Status på implementation:</b>	<b>13</b>
<b>Proces:</b>	<b>14</b>
<b>Bilag:</b>	<b>15</b>

## Indledning:

Cupcake projektet er en demo på hvordan en hjemmeside for den fiktive virksomhed *Olsker Cupcakes* kan fungere og se ud. Formålet med projektet er først og fremmest at skabe en brugbar hjemmeside, hvortil udseendet kommer i anden række.

Under afsnittet teknologivalg er der angivet hvilke programmer der benyttes i projektet, men primært er der tale om Java og HTML med brug af bootstrap og tomcat som containeren.

## Baggrund:

*“Vi har landet en vigtig opgave fra Olsker Cupcakes. Det er endnu et dybdeøkologisk iværksættelseventyr 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. En mockup er en meget løs skitse, som viser hvordan det færdige website skal se ud. Det er selvfølgelig ikke alt som er med, og som er tænkt igennem, så det er vores opgave at stille spørgsmål til manglende funktionalitet, komme med forslag osv.*

*Når man som leverandør skal løse sådan en opgave, er det godt at dele opgaven op i små etaper. Så tænk fra starten over hvor lidt vi kan lave for at få den første prototype i luften. Med andre ord: vi skal ikke lade os forblænde af hipsternes farver og striber og mange funktionaliteter. Vi æder elefanten lidt ad gangen.”*

- Citat <https://cphbusiness.mrooms.net/mod/assign/view.php?id=324084>

## Teknologivalg:

Her kan man læse om de forskellige teknologier, som vi har valgt at bruge i denne opgave.

Til at oprette en database brugte vi **Mysql 8.0.22 workbench** til at lave en **Mysql 8.0.22 Server** der ville indeholde de forskellige værdier, som vi mente var nødvendige.

Vi brugte **paint.net 4.2.15 (Final 4.215.7694.37221)** til at lave mockups, som vi baserer vores forskellige sider på.

Vi lavede forskellige objekter og andet server relateret i **Java 1.8 (java 8)**.

Vi brugte **IntelliJ IDEA 2021.1 (Ultimate Edition)(Build #IU-211.6693.111, built on April 6, 2021)** som vores code editor til at arbejde i. De forskellige kodesprog som vi brugte, var **Javascript**, til vores server relateret indhold og de forskellige objekter, mens vi brugte **HTML5** til at opsætte de forskellige sider, som brugeren vil navigere.

Vi brugte **Tomcat 9.0.44** og **Tomcat 9.0.45** til at opsætte en lokal server så vi kunne teste vores produkt.

# Krav:

Her kan man læse om de forskellige userstories som blev udgivet sammen med opgaven.

## **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 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 ordrelinier 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 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:**

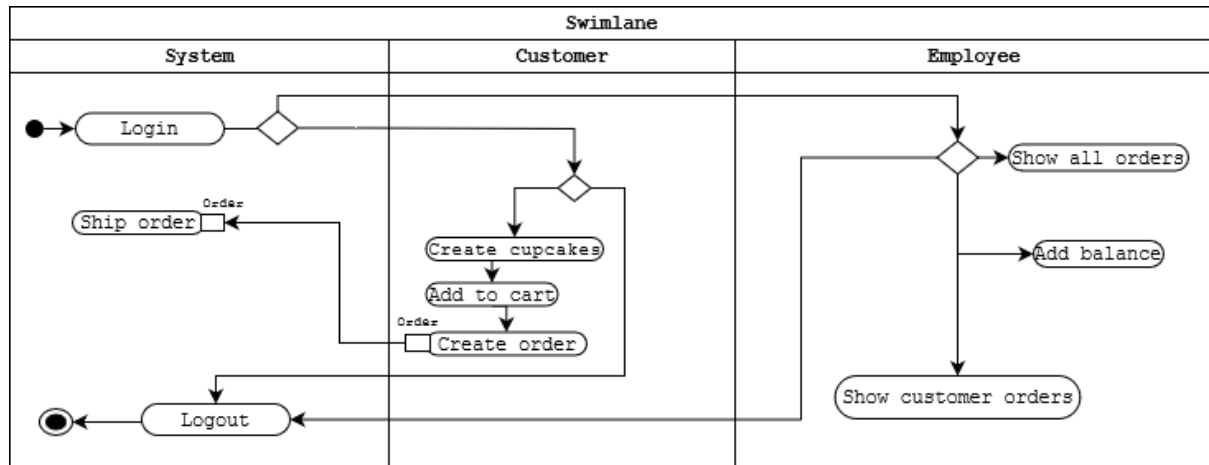
Som kunde kan jeg fjerne en ordre fra min indkøbskurv, så jeg kan justere min ordre.

## **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:

Her vil vi beskrive hvordan vi forventer at forløbet foregår for en bruger.



Systemet ser på om det er en kunde(customer) eller en ansat(employee) som logger ind.

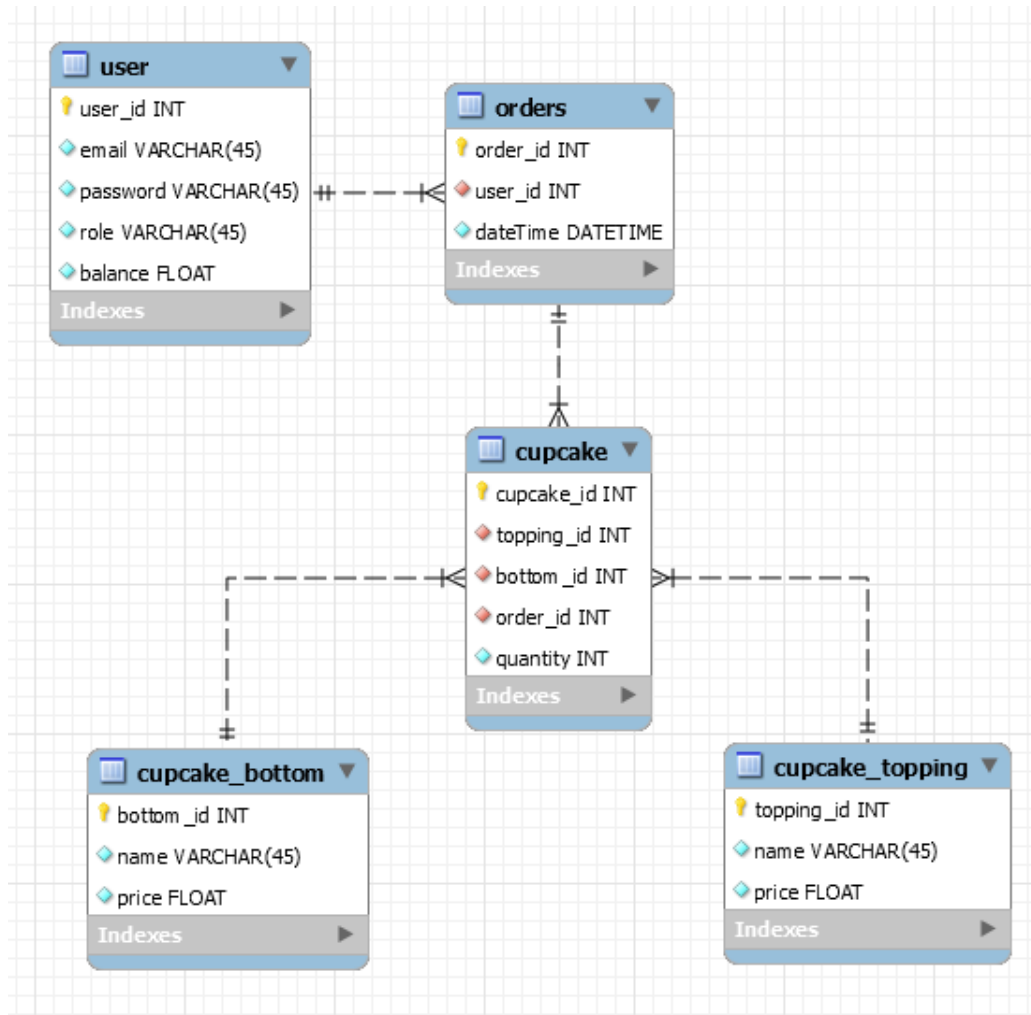
Derefter hvis det er en kunde som har logget ind, så har de mulighed for at lave cupcakes(create cupcakes) som de derefter kan sætte ind i deres kurv(add to cart) og til sidst bestille(create order). Eller vælge at logge ud og slutte deres session.

Derimod hvis det er en ansat som er logget ind i stedet for er der andre valgmuligheder.

En ansat kan vælge mellem "Show all orders", "Add balance", "Show customer orders" eller logge ud.

## EER diagram:

Her vil vi beskrive hvordan vi har opsat vores database.



Som en bruger(user) så kan man lave flere ordre(orders), derfor er der en én til mange relation mellem user og orders. En ordre kan have mange cupcakes, hvor man identificerer ved hjælp af order\_id til at vide hvilken ordre hver cupcake tilhører.

En cupcake består af en “bottom” og en “topping” som vist på billedet ovenover(cupcake\_bottom & cupcake\_topping). Hver “bottom” og “topping” kan blive brugt til flere forskellige unikke cupcakes så derfor har begge af dem en én til mange relation med cupcake.



# Domæne model:

## **Version 1:**

Den første version af vores domæne model har serveren i centrum. Den har en database som den henter information fra, samt flere brugere. Brugeren indeholde ind rolle og en shopping kurv, som har en liste over de forskellige produkter man kan købe, og brugeren indeholder også en ordre historie. Ordre historien indeholder de tidligere købte produkter. Produktet er opsat af en top og en bund. (Bilag 2a)

## **Final version:**

Denne version er ovenstående det samme, men den har dog nogle ændringer som stammer fra vores valg, som vi lavede da vi kodede projektet. Der er blevet tilføjet to nye enums der bliver brugt til at opsætte cupcakesne. Item indeholder nu kun de to nye enums og Cart har ikke længere en arraylist og har i stedet et hashmap, som også indeholder antallet af produktet som brugeren har valgt. (Bilag 2b)

# Mockup:

**Bilag 1a:** Viser et potentielt billede af en kundes indkøbskurv. Herfra kan kunden ændre på antallet af de sammensætninger der er blevet valgt og til sidst bekræfte sin bestilling ved at trykke “submit”. I sidste ende blev funktionaliteten ændret en smule, da vi fjernede det trin som er vist her og man i stedet kan tilføje og ændre direkte på en bestillingsside.

**Bilag 1b:** Viser et billede af kundens “Mine ordrer” side, hvor man som kunde kan se alle tidligere bestillinger og via en “show” knap kan få flere detaljer om sammensætningen af cupcakes, antal og pris.

**Bilag 1c:** Her fremgår de detaljer kunden ville kunne se ved at trykke “show” på “Mine ordrer” siden fra bilag 1b.

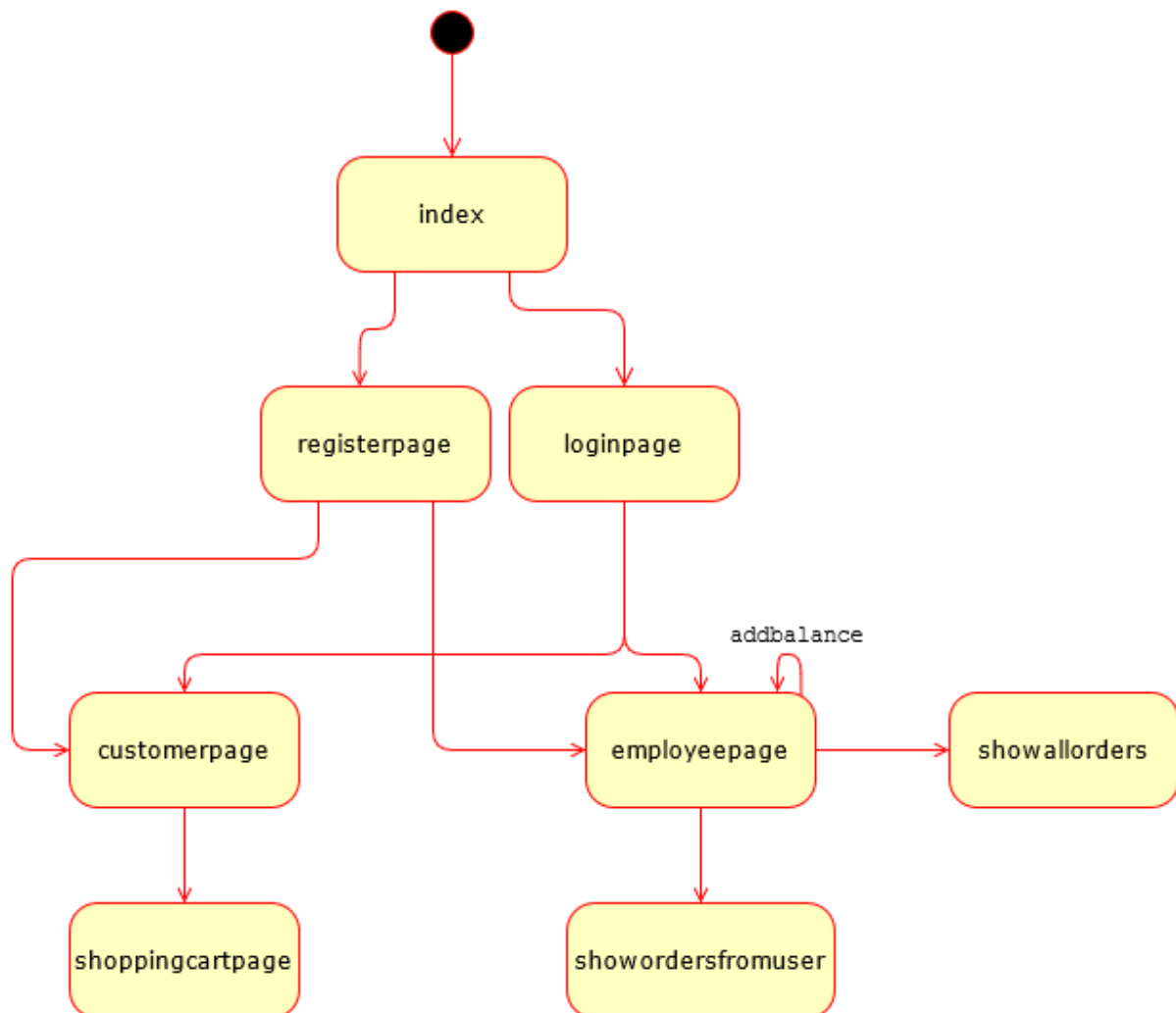
**Bilag 1d:** Viser en admins side, hvor der er mulighed for at se alle kunder, se alle ordrer eller søge efter en specifik kunde eller ordrer. I dette tilfælde er det vist hvordan en oversigt over kunderne er valgt og vores admin kan trykke “Se ordrer” for at få en oversigt over den enkelte kundes tidligere bestillinger.

**Bilag 1e:** Her har vores admin valgt at klikke “Se ordrer” for kunden Robin på den foregående side vist i bilag 1d. Vores admin får samme oversigt, som Robin selv vil kunne finde, vist i bilag 1b, med mulighed for igen at klikke “show” og få en mere detaljeret oversigt for den enkelte ordrer.

**Bilag 1f:** Viser en oversigt over alle ordrer, hvis man som admin vælger at søge efter ordrer fremfor kunder. Herfra kan vores admin igen klikke “show” for flere detaljer.

# Navigationsdiagram:

Her vil vi vise og beskrive hvordan man kan navigere gennem vores produkt.



Brugeren start med at først at komme til index siden, som giver brugeren mulighed for at navigere til register page og login page.

På register page vil sende den sende brugeren videre ud fra den rolle den nylig registrerede brugere har og det samme gælder for login page, som dog i stedet brugere allerede eksisterende brugere.

Fra customer page kan man komme videre til brugerens shopping kurv, mens fra employee page kan man komme til "show all orders" og "show orders from user".

## Særlige forhold:

### **Information:**

Igennem hele sessionen er der nogle forskellige informationer, som er nødvendige at gemme, for at gøre nogle funktioner muligt for brugeren.

De information som bliver gemt i session er information omkring brugeren, omkring indkøbskurven og omkring priserne.

#### **- User information:**

Den gemte information omkring brugeren indeholder brugerens email og brugerens rolle.

#### **- Shopping cart information:**

Den gemte information omkring brugerens shopping kurv indeholder en instance af et kurv object, som indeholder de forskellige cupcakes, som brugeren har tilføjet.

#### **- Price information:**

Den gemte information omkring priserne indeholder priser for de forskellige toppe og de forskellige bunde, som en cupcake kan blive lavet af.

### **Exceptions:**

Selve application håndtere exceptions, som der kan opstå gennem hele programmet, ved at sende brugeren til en error page, som vil udskrive exceptionel, samt et nummer der beskriver fejlen.

## Status på implementation:

I denne del af rapporten, vil man kunne læse om hvordan vi har implementeret de forskellige userstories, som blev udgivet sammen med opgaven.

### **US-1:**

Man kan oprette og sende en cupcake med hvor mange man vil have af den, men der er ikke blevet opsat en betal knap.

### **US-2:**

Man kan oprette en ny bruger gennem en sign up side. Den kræver en email(dog ikke en virkelig en), og at man indtaster password to gange, hvor de skal matche hinanden. Den oprettede profil vil altid have rollen "customer".

### **US-3:**

Ved at først logge ind på en bruger, som har rollen "employee", kan man derefter gå til en side hvor man kan se alle brugere med rollen "customer". På denne side kan man indsætte et beløb på en bruger af gangen.

### **US-4:**

Når man er logget ind som kunde, kan man se alle valgte cupcakes i sin kurv, samt priser.

### **US-5:**

Når man er logget ind kan se brugerens email på alle sider.

### **US-6:**

Når man er logget ind som admin, kan man vælge en bruger og se deres tidligere bestillinger.

### **US-7:**

Når man er logget ind som admin, kan man se alle kunder i systemet og deres bestillinger.

### **US-8:**

Ved at sætte antallet af en cupcake type i ens kurv, vil systemet automatisk fjerne den efter den næste opdatering.

### **US-9:**

Denne blev ikke implementeret.

## Proces:

Vi startede projektet med at gennemgå den startkode vi havde fået udleveret. Herfra lavede vi i fællesskab nogle små ændringer til koden, da et vigtigt punkt var at beholde det samme design på hver side, hvortil vi benyttede en generisk side, som alle andre sider nedarvede fra.


Herefter var vores plan at fordele arbejdet ved hjælp af et kanban board (Bilag 3a), hvor hver deltager valgte en user story med henblik på at merge de forskellige branches og finpudse eventuelle mangler/konflikter til sidst i forløbet. En stor del af startkoden havde allerede taget højde for de user stories beskrevet i kravene US-2 og US-5.

I praksis oplevede vi et skel mellem de individuelle færdighedsniveauer blandt medlemmerne, så det var nødvendigt at få samlet medlemmer til at kigge på en løsning til US-1.

Alle medlemmer var aktive og produktive efter deres færdighedsniveau. Igennem forløbet var der god kommunikation og gruppemedlemmerne var gode til at løfte i fællesskab og finde løsninger til diverse problemer, som vi oplevede med de individuelle processer.

# Bilag:

1a:



Order







Kunder

joker@gotham.com

3 


topping	bottom	pris	Antal
chocolate	chocolate	10	<input type="text" value="1"/>
vanilla	raspberry	12	<input type="text" value="2"/>
strawberry	orange	15	<input type="text" value="5"/>
		37	<input type="button" value="submit"/>

1b:



Kunder

joker@gotham.com

3 

Dato	2020-12-21	Antal	8	total pris	37	<input type="button" value="show"/>
Dato	2020-12-21	Antal	8	total pris	37	<input type="button" value="show"/>
Dato	2020-12-21	Antal	8	total pris	37	<input type="button" value="show"/>

1c:




Order
Kunder
joker@gotham.com
3


topping	bottom	pris	Antal
chocolate	chocolate	10	1
vanilla	raspberry	12	2
strawberry	orange	15	5
		37	8

1d:




Order
Kunder
joker@gotham.com
3


Se alle ordrer
søg

Bruger	
joker@gotham.com	Se ordrer
robin@gotham.com	Se ordrer
Bobby@supernaturalIndustries.com	Se ordrer

1e:






Order

Kunder

joker@gotham.com

3 

Robins ordrer

Dato	2020-12-21	Antal	8	total pris	37	<div>show</div>
Dato	2020-12-21	Antal	8	total pris	37	<div>show</div>
Dato	2020-12-21	Antal	8	total pris	37	<div>show</div>

1f:




Order

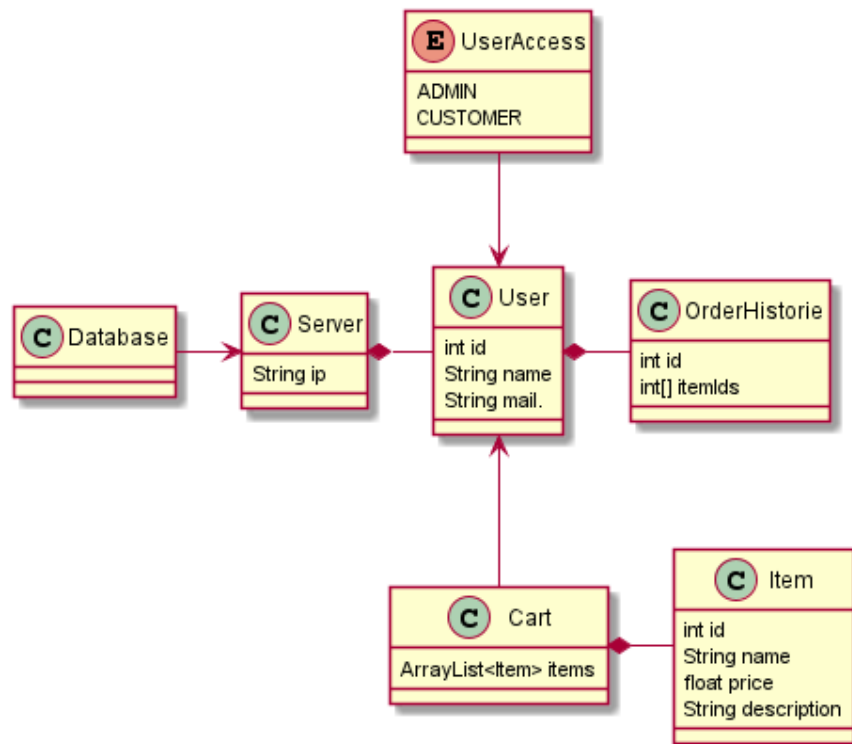
Kunder

joker@gotham.com

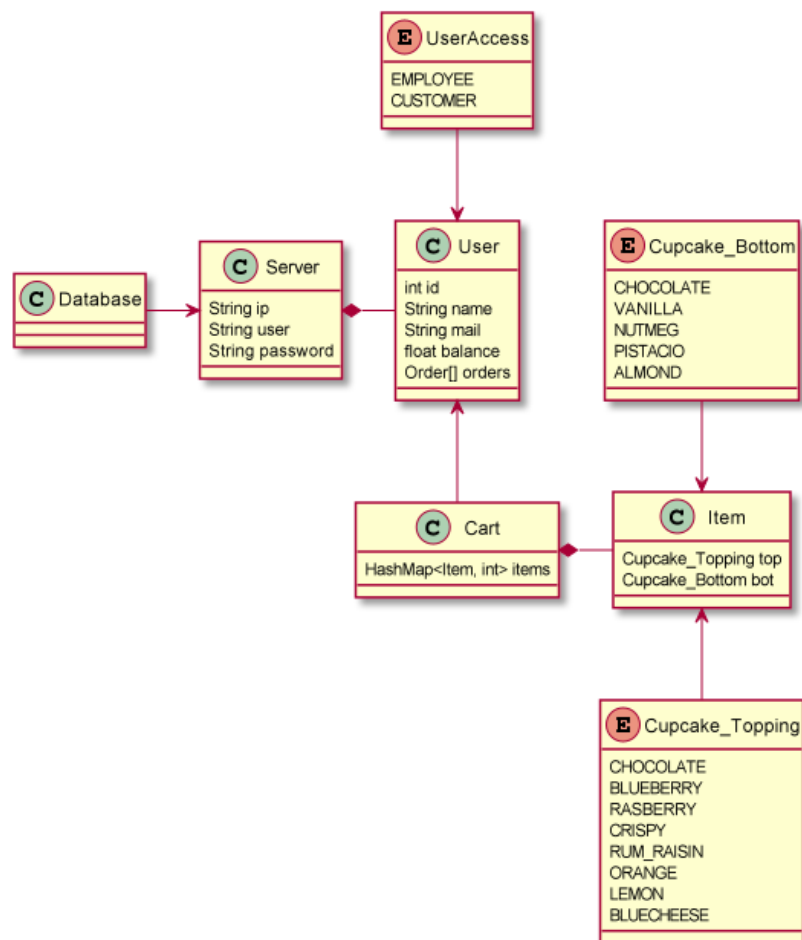
3 

Bruger	Dato	2020-12-21	Antal	8	total pris	37	<div>show</div>
joker@gotham.com	Dato	2020-12-21	Antal	8	total pris	37	<div>show</div>
robin@gotham.com	Dato	2020-12-21	Antal	8	total pris	37	<div>show</div>

2a:



2b:



3a:

Userstories (funktionelle krav)  
Updated 4 days ago

1 To do	1 In progress	7 Done
<div>US-9 #9 opened by ChristofferQ</div>	<div>US-1 #1 opened by ChristofferQ</div>	<div>US-2 #2 opened by ChristofferQ</div>
		<div>US-5 #5 opened by ChristofferQ</div>
		<div>US-6 #6 opened by ChristofferQ</div>
		<div>US-7 #7 opened by ChristofferQ</div>
		<div>US-3 #3 opened by ChristofferQ</div>
		<div>US-4 #4 opened by ChristofferQ</div>
		<div>US-8 #8 opened by ChristofferQ</div>