

## Olsker Cupcakes E-klassen, gruppe 750:

- Yones, [cph-ye7@cphbusiness.dk](mailto:cph-ye7@cphbusiness.dk), Github: Youneselb
- Christoffer, [cph-cf161@cphbusiness.dk](mailto:cph-cf161@cphbusiness.dk), Github: Foldager94
- Ali, [cph-at206@cphbusiness.dk](mailto:cph-at206@cphbusiness.dk), Github: AliTahir78
- Frederik, [cph-fj111@cphbusiness.dk](mailto:cph-fj111@cphbusiness.dk), Github: TheRiisager

<b>Indledning:</b>	<b>3</b>
<b>Interessenter:</b>	<b>3</b>
<b>Baggrund:</b>	<b>4</b>
<b>Teknologivalg:</b>	<b>4</b>
<b>Krav</b>	<b>5</b>
<b>Domæne model</b>	<b>5</b>
<b>Overblik over layer:</b>	<b>6</b>
<b>ER diagram:</b>	<b>8</b>
<b>Navigationsdiagram:</b>	<b>9</b>
<b>Sekvensdiagram:</b>	<b>10</b>
<b>Aktivitetsdiagram:</b>	<b>11</b>
<b>Særlige forhold</b>	<b>11</b>
<b>Status på implementation:</b>	<b>12</b>

## Indledning:

Fokus for rapporten og cupcake projektet, er opbygningen af en simpel webshop løsning, som skal gøre det muligt for Olsker Cupcakes' kunder, at foretage bestillinger efter eget valg, som efterfølgende kan afhentes i butikken. Olsker Cupcakes har derfor en række forestillinger om krav til opbygningen af hjemmesiden.

## Interessenter:

- Ejer
- Admin
- Ekspedient/konditor
- Kunder
- Revisor (inddraget som eksempel)
- Rengøringspersonale (inddraget som eksempel)

### - Roller:

	Ikke indflydelse på projektet	Indflydelse på projektet
Påvirket af projektet	Ekspedient/konditor Kunder	Ejer Admin
Ikke påvirket af projektet		

(Revisor og rengøringspersonalet, er ikke tildelt roller, da det ikke er relevant for projektet. Dog kunne man sætte dem i boksen "Ikke påvirket af projektet/Ikke indflydelse på projektet")

## Inddragelse af interessenter:

**Ejer:** Ejeren af virksomheden, er den største interessent i forhold til dette projekt. Det er derfor vigtigt at sikre, at det vi udvikler, er afstemt med ejerens forventninger til hjemmesidens funktionalitet. Ejer skal derfor inddrages ved løbende kort møder med projektleder, i udviklingsprocessen. Der skal være afstemning med ejer før påbegyndelse af alle opgaver, samt efter opgavernes afslutning, for at sikre afstemmelse af opgavens indhold og resultater.

**Admin:** Admin er en vigtig interessent, da det er ham som skal drifte hjemmesiden efter projektets afslutning. Admin skal inddrages i projektet, så det sikres at vi leverer den funktionalitet han har behov for, inden for de økonomiske og tekniske rammer.

**Ekspedient/konditor:** Det er ikke planlagt at inddrage ekspedienten i udviklingsprocessen, da vi antager at ejeren fuldt ud kender de arbejdsopgaver ekspedienten udfører.

Vi inddrager ekspedienten i test af bestillingssystemet.

**Kunder:** Inddrages ikke i udviklingsprojektet. Dette kunne være gjort ved at ejeren på et tidligt tidspunkt eks. have lagt et spørgeskema i butikken, som kunne belyse kundernes ønsker til hjemmesidens funktionalitet.

Hvis der er kunder som ejeren har et særligt godt forhold, kan disse inddrages i test af hjemmesiden, så det sikres, at vi har udviklet en hjemmeside som kunderne let og intuitivt kan anvende.

## **Baggrund:**

Olsker Cupcakes er et iværksætterprojekt, som i sin enkelthed består af en lille forretning, der bager cupcakes, hvor toppe og bunde kan designes af den enkelte kunde og bestilles over en webshop til afhentning.

Valget af cupcakes tager afsæt i et fast sortiment, der består af 5 bunde og 9 toppe, som den enkelte kunde selv sammensætter efter lyst.

Olsker cupcakes har bedt en leverandør, at bygge en webshop som opfylder dette krav udefra en mockup af, hvordan de forestiller sig, hjemmesiden skal se ud. Udover forestillinger om designet af webshop, har forretningen en række krav til hjemmesidens funktionalitet, som udmunder sig i en række userstories, der dækker over, hvad brugeren/kunden skal kunne foretage sig på hjemmesiden, og hvad administratoren har behov for at kunne foretage sig.

Grundlæggende set skal en kunde kunne registrere sig som bruger på siden, vælge top og bunde fra sortimentet, foretage en bestilling, og betale via kredit eller ved afhentning i butikken. Både administrator og kunden, skal have muligheden for at få overblik over ordrer og bestillinger.

## **Teknologivalg:**

Intellij 8.1 er brugt til kodningen af webshoppen. Alt relateret til opbygningen af databasen (tabellen, data osv.), er lavet i MySQL Workbench. Forbindelsen til serveren sker via Tomcat og Digital Ocean bruges som droplet.

Til at designe mockups af hjemmesidens visuelle udtryk, har vi brugt Adobe XD, samt illustrator til eksempelvis redigering af baggrundsbillede osv.

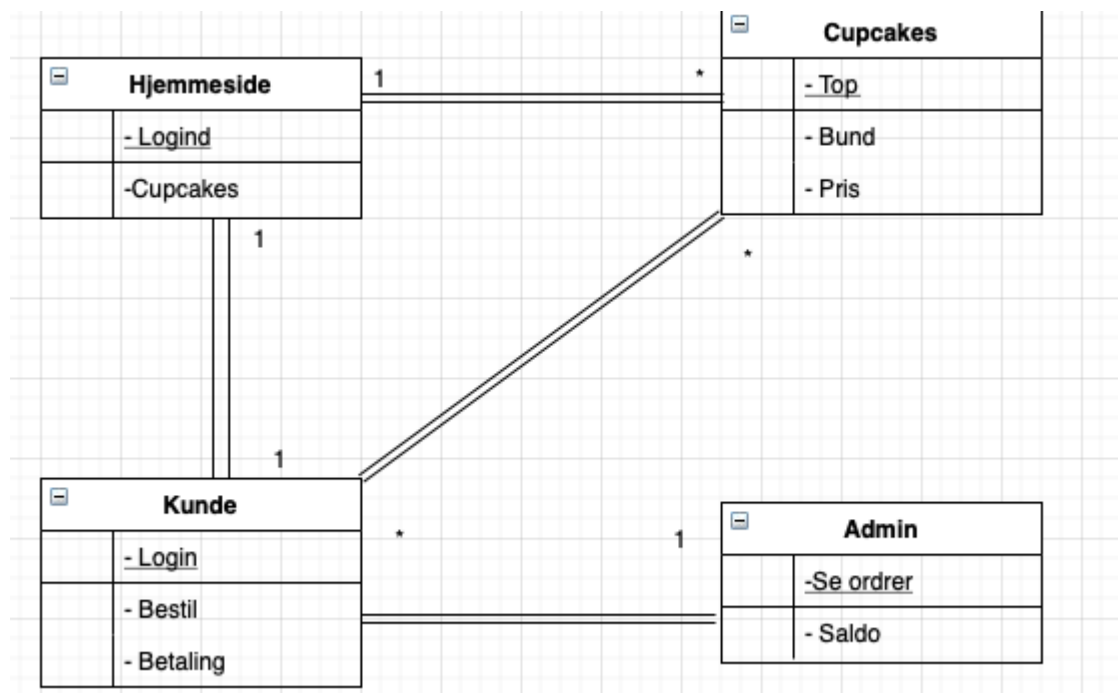
## Krav

Forretningens vision med webshop løsningen, er ønsket om at tiltrække flere kunder, ved et simpelt og enkelt system til bestilling. Medarbejderne skal derfor blot koncentrere sig om bagning, og dermed bliver arbejdsprocesserne effektiviseret og tidsbesparende. Kunderne får et bedre overblik over forretningens sortiment, og kan i ro og mag, overveje sammensætninger, før deres bestilling.

For at realisere visionen, har virksomheden en række user-stories, der skal gøre sig gældende:

1. Kunden skal kunne sammensætte top og bund på cupcakes via et dropdown menu, i en personlig kundeside, for derefter at afhente i butikken.
2. Kunden skal kunne oprette en bruger på hjemmesiden, for at kunne betale og gemme ordrer.
3. Administratoren skal kunne indsætte penge på kundernes saldo(via MySql), som gør det muligt for kunderne at foretage bestillinger.
4. Kunden skal kunne se sin ordrerlinier, for fuldt overblik over samlet pris.
5. Siden skal vise brugernes e-mail på alle sider, når de er logget på.
6. Administratoren skal have mulighed for at se alle ordrer, så der er overblik over alle bestillinger.

## Domæne model



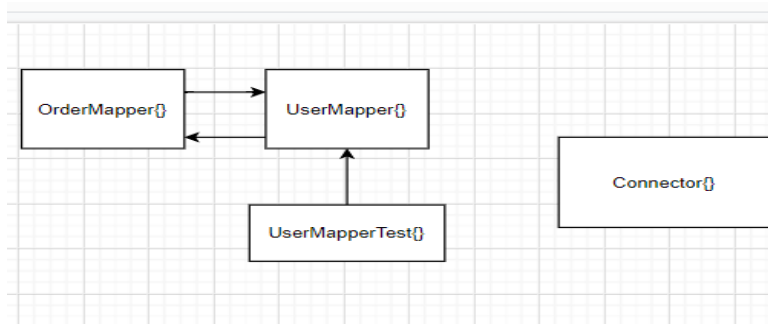
Domænemodellen giver et forsimpelt billede af sammenhængen mellem de forskellige entiteter og kardinaliteten imellem dem.

## Overblik over layer:

### DbConnector:

Connector klassen er ikke forbundet til andet, da dens eneste funktion er at skabe forbindelse til databasen og projektet. UserMapper test er forbundet til UserMapper, da dens eneste funktion er at sikre om alt i klassen, fungerer som det skal.

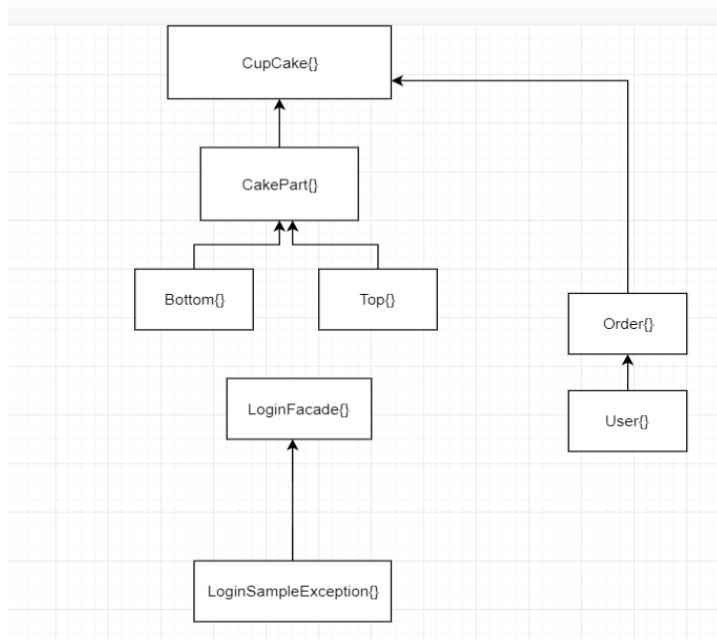
OrderMapper og UserMapper er forbundet til hinanden, da en bruger godt kan have en ordre i en 1 til mange relation, men en ordre kan ikke have flere brugere, hvorimod flere brugere kan have mange ordrer.



### Function Layer:

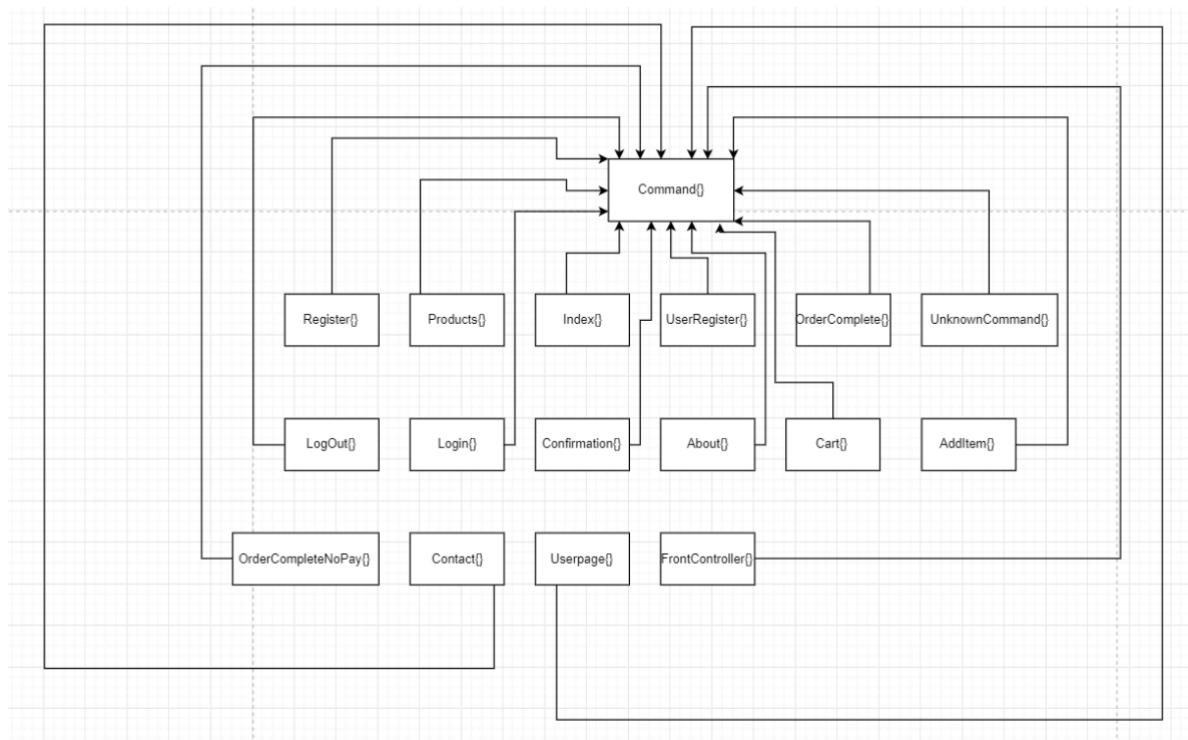
Funktion layer indeholder forskellige relationer til cupcake{} øverst på diagrammet. Det skyldes at cakepart{}, er sammensætningen af top{} og bottom {}, hvor hele cupcaken bliver gemt og senere kan bestilles. User{} er forbundet til order{} med en 1 til mange relation, da en bruger godt kan have mange ordrer, hvorimod en ordre ikke kan have mange brugere, grundet unikke autogenerated ID'er. Order{} bliver så sendt videre til cupcake klassen, som sætter cakepart{} og order{} sammen, for at bestillingen kan foretages og sikre, at der er penge nok på saldoen.

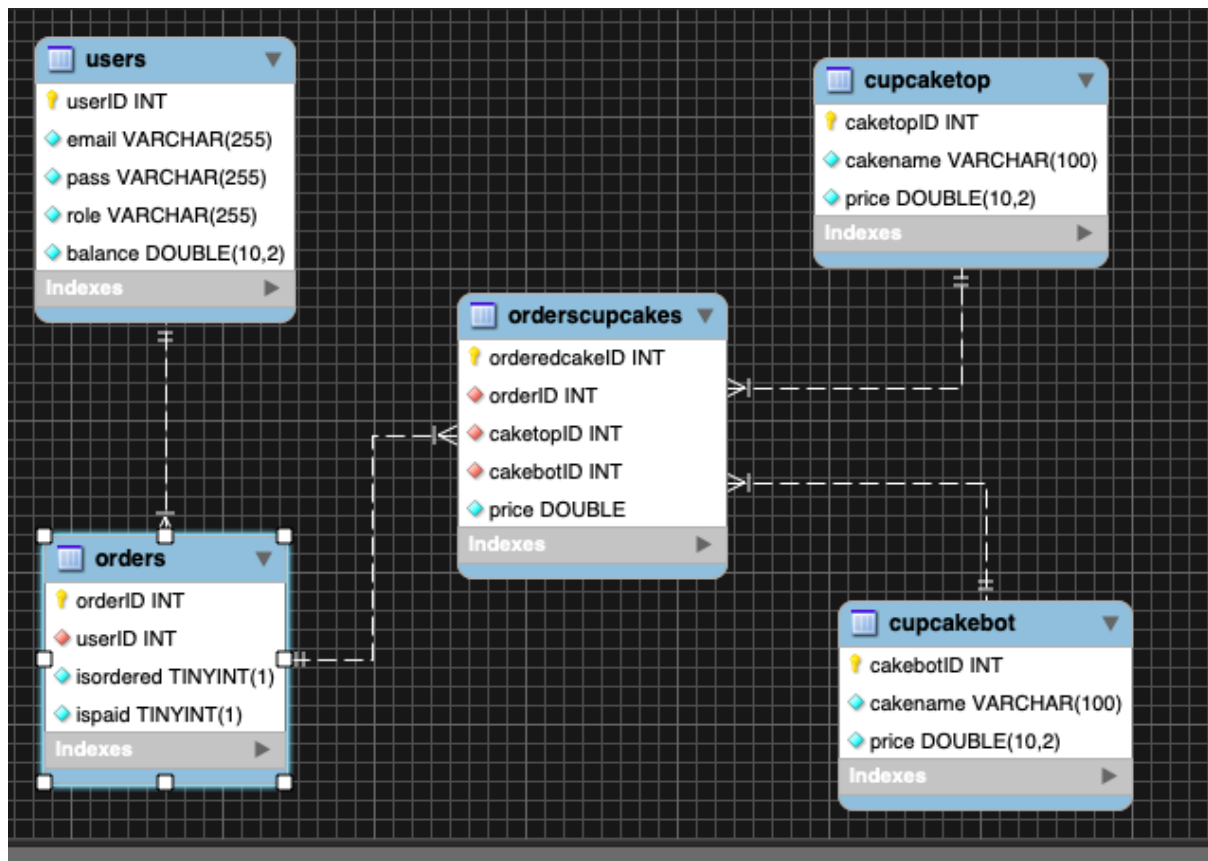
LoginSampleException{} er forbundet til LoginFacade{}, da LoginFacade{}, sørger for at man indsætter en email og kode, som findes i databasen, hvorefter man bliver videresendt til admin eller kunde siden. LoginSampleException{} sikrer, at emailen eksisterer i databasen, og sender brugeren tilbage til index siden, hvis det ikke er tilfældet.



### Presentation Layer:

Alle klasser er forbundet til command klassen, da den har indflydelse på alle interaktioner med hjemmesiden. I command klassen bliver alle klasser instantieret so objekter, og derfor er alle klasser tilknyttet command{}

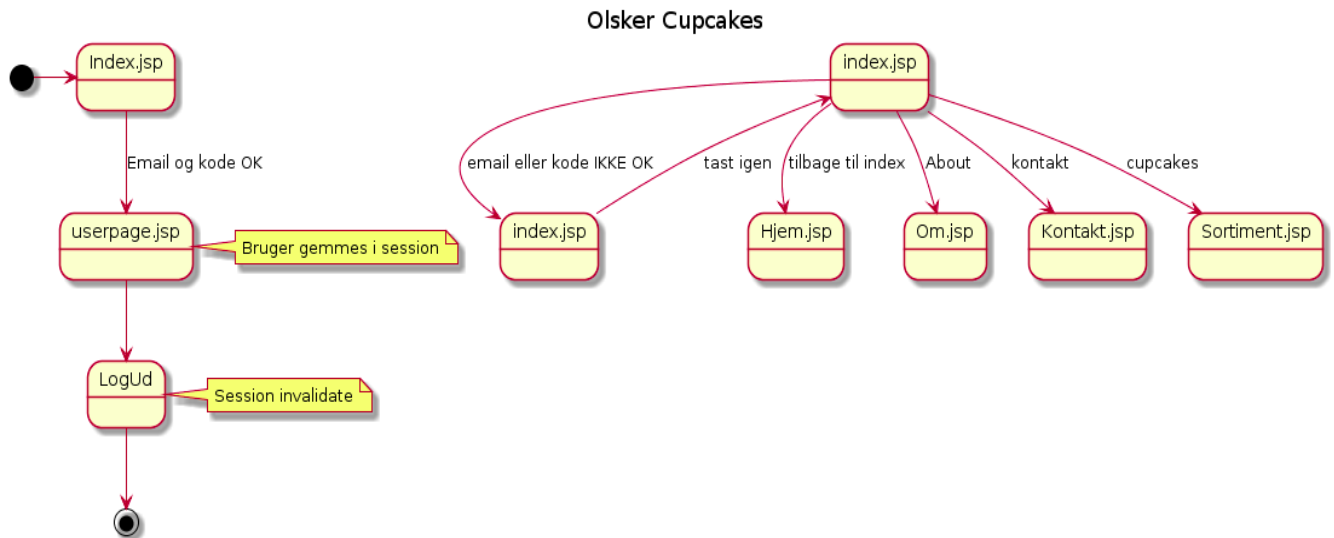


**ER diagram:**

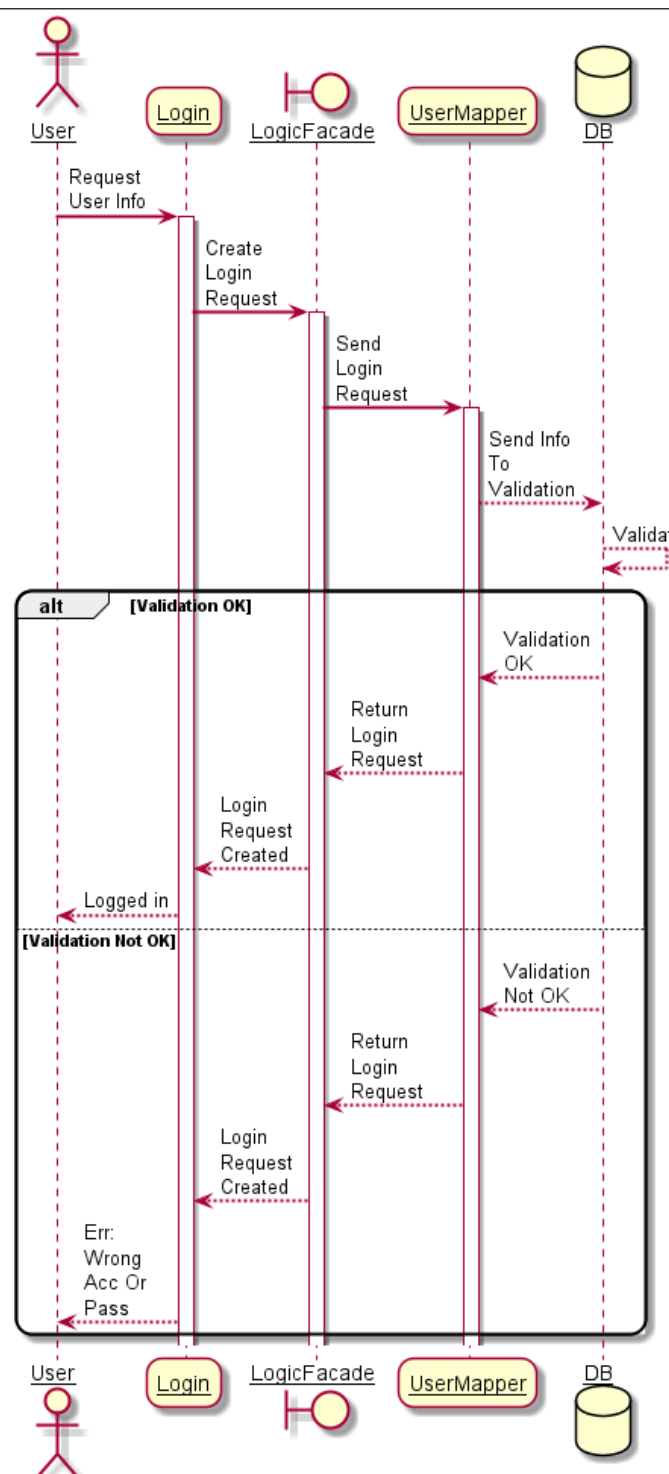
ER diagrammet har 5 tabeller, med en unik nøgle ID, der bliver automatisk genereret.



## Navigationsdiagram:



Navigations diagrammet giver et overblik over navigationen på hele hjemmesiden. Her er `index.jsp` siden udgangspunktet for det hele, hvor man så bevæger sig rundt til de andre `jsp`-sider (om, kontakt, sortiment og kundesiden). Da hjemmesiden er rimelig overskuelig, er der ikke behov for, at diagrammet viser hvordan en entitet overgår til en anden tilstand, da det primært er login-siden, der er relevant i den forbindelse, hvor man enten er logget ind eller ud. Her kunne man have lavet et "composite state", for at demonstrere at man er logget på eller af.

**Sekvensdiagram:**

Sekvensdiagrammet viser hvordan en bruger bliver logget ind i systemet. Brugeren taster navn og password på index.jsp siden, som så sender brugeren videre i systemet.

## Aktivitetsdiagram:



Vi har valgt et aktivitetsdiagram, der fokuserer på hovedformålet med hele hjemmesiden og den grundlæggende aktivitet/handling, som er i fokus: bestilling af cupcakes. Det er ikke et særlig kompliceret scenarie med mange forgreninger, og fungerer derfor på samme måde som de fleste andre bestillingsløsninger.

Når man er logget ind, vælger man top og bund fra dropdown menuen, hvorefter man kan vælge at betale fra ens saldo for at færdiggøre ordren eller afhente i butikken. Dette er det eneste sted i processen hvor der kan opstå et *fork*, men da de fører til samme resultat på hjemmesiden, er forgreningen ikke illustreret i diagrammet.

## Særlige forhold

- Sessions bliver gemt ved brug af vores user objekt. User objektet indeholder en rolle (user eller admin), en email-adresse, password og en ordre. Balancen/saldo er også tilknyttet enhver bruger, som afgør om de kan bestille cupcakes, men balancen hentes fra userMapper og er derfor ikke en del af user objektet.
- Den eneste validering vi har, er hvorvidt email og kodeord findes i databasen, som så gør det muligt at logge ind på kundesiden. Man kunne også have implementeret en "opret bruger" funktion, der valideret om en email adresse allerede var i brug, og hvis ikke, så kunne den gå ind og oprette brugen på en specifik mailadresse.
- Vi har ikke implementeret noget sikkerhed omkring login, udover validering af brugeroplysninger.

- Admin kan ikke foretage de samme handlinger på hjemmesiden, som den almindelig bruger/kunde kan. Derfor kan admin ikke bestille cupcakes, men kun få et overblik over ordrer, saldo osv.
- Indkøbskurven validerer om en bruger har en ordre der er "0" i isordered, hvorefter den trækker indkøbskurven ind og viser de pågældende data i ordren. Hvis alle ordrer har "1" i isordered, opretter den en ny ordre række i databasen.

### **Status på implementation:**

- Sortiment siden mangler overblik over cupcakes, men bliver vist som dropdown menu i I kundesiden.
- Alle siderne på hjemmesiden er stylet.
- Vi har ikke opdaget fejl på hjemmesiden, men der mangler funktionalitet til at fjerne cupcake ordre, hvis en bruger f. eks har tilføjet flere cupcakes til en enkelt ordre.
- I forhold til CRUD, gør vi ikke meget ud at slette eller fjerne indhold i databasen/tabellerne.