

Medlemmer:

Albert Løhde cph-al324@cphbusiness.dk <https://github.com/AlbertSylv>

Daniel Bisgaard cph-db144@cphbusiness.dk <https://github.com/DanielBisgaard>

Artem Ivanov cph-ai62@cphbusiness.dk <https://github.com/AIcode97>

Klasse: D

Uddannelse: Datamatiker

(cupcake projekt ligger på denne github: <https://github.com/DanielBisgaard/Cupcake>)

Datamatiker 2020

Olskerscupcake

Opgave



Introduktion

Udgangspunktet i opgaven er at opsætte en hjemmeside til Olsker Cupcakes til bestilling og betaling af deres cupcakes. Formålet med opgaven er at kunne skabe en funktionel hjemmeside hvor brugeren kan registrere sig på og derefter logge ind på siden for at lave en bestilling.

Baggrund:

Olsker Cupcakes er et økologisk cupcake firma fra bornholm som har ramt den helt rigtige opskrift indenfor cupcake verdenen. De står for produktion af en række lækre cupcakes som de vil dele med deres kunder.

Olsker Cupcakes vil have en hjemmeside lavet for at kunne betjene deres kunder hurtigt og effektivt, det skal være let for deres kunder at bestille fra hjemmesiden og kunne hente cupcakes fra den fysiske butik.

Firmaet har krævet at en kunde kan registrere sig på hjemmesiden og logge ind for at lave en bestilling. Bestillingen foregår ved at kunden selv skal opbygge sin cupcake ved at vælge top og bund fra det udvalgte menu på hjemmesiden og derefter kan checke ud gennem indkøbskurven.

Teknologivalg:

I dette projekt har vi valgt at udvikle vores produkt i en kombination af specifikke programmer.

Front-end / Back-end:

front-end er der blevet brugt en Bootstrap4 navigation bar m. en login -og registrerings dropdown menu..

Programming software: Intellij DEA 2019.3.3 er programmet vi har brugt til at kode front-end og back-end af vores hjemmesiden.

Project SDK: Java version 1.8 er versionen af java som vi har brugt til at kode back-end delen af hjemmesiden

Project language level: Lambdas er det valgte sprog for vores projekt.

Database software: MySQL Workbench 8.0 CE er det valgte database værktøj for opsætningen af database i projektet.

GitHub.com: GitHub er brugt i projektet for en nem samarbejde og udvikling af projektet i et hold til at kunne nemt dele, hente og sammenkoble forskellige gruppemedlemmers dele af projektet. Som ekstra "hjælpemiddel" har vi brugt Github Desktop til at lave et præcist opdaterings mønster i forhold til at den sørger for at der ikke bliver pushet før der er blevet pullet - Så det har sparet en masse tid.

Server: Tomcat 8.5.51 har været den valgte type af server setup for at køre projektet i et browser miljø.

Krav:

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 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.

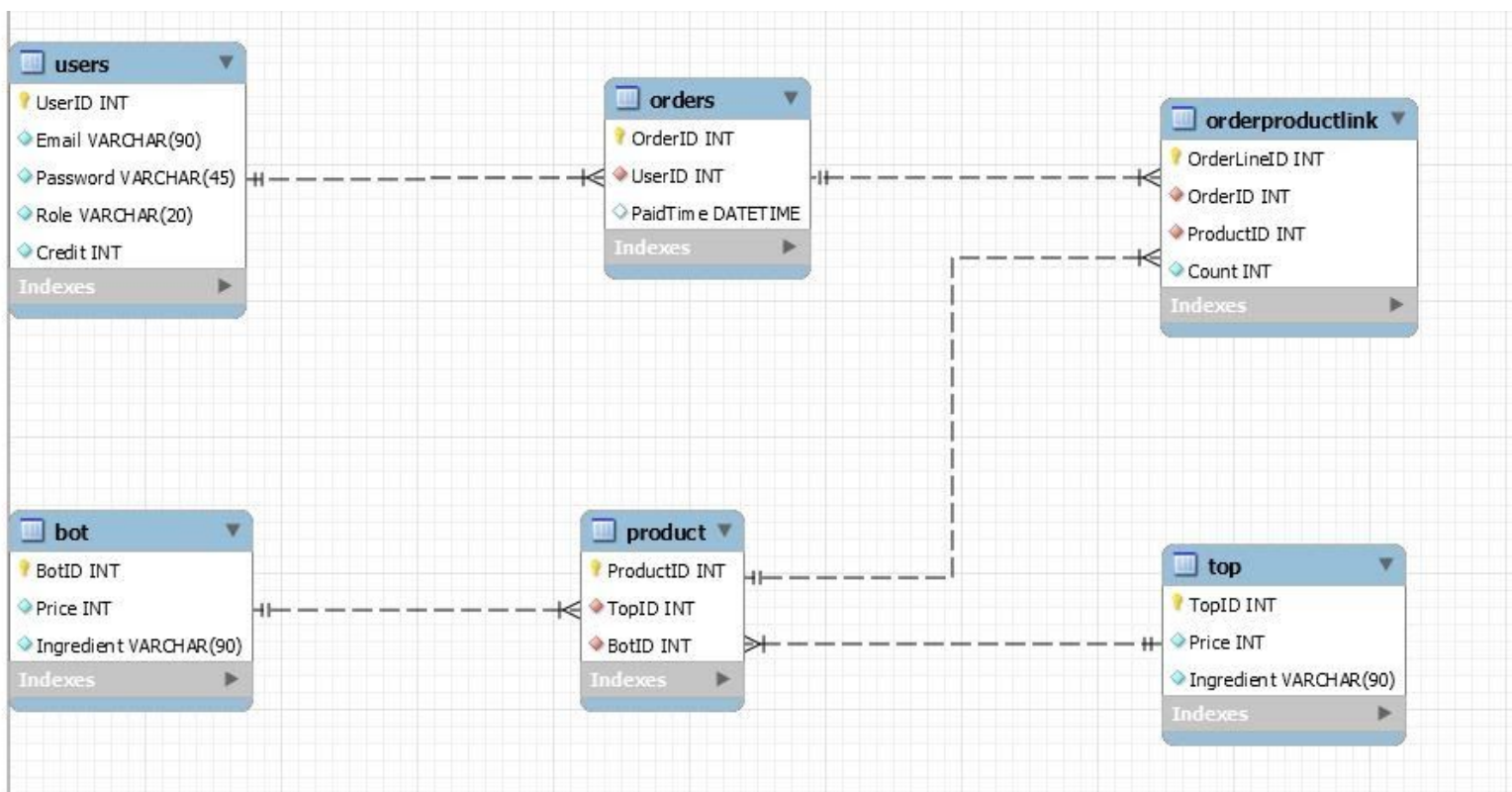
US-7: Som administrator kan jeg se alle kunder i systemet og deres ordrer, sådan at jeg kan følge op på ordrer og holde styr på mine kunder.

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.

ER-Diagram:

ER diagrammet er en repræsentation af vores database, hvordan den er koblet sammen med primary keys og hvilket indhold de forskellige tables har. Den er normaliseret på tredje normalform.



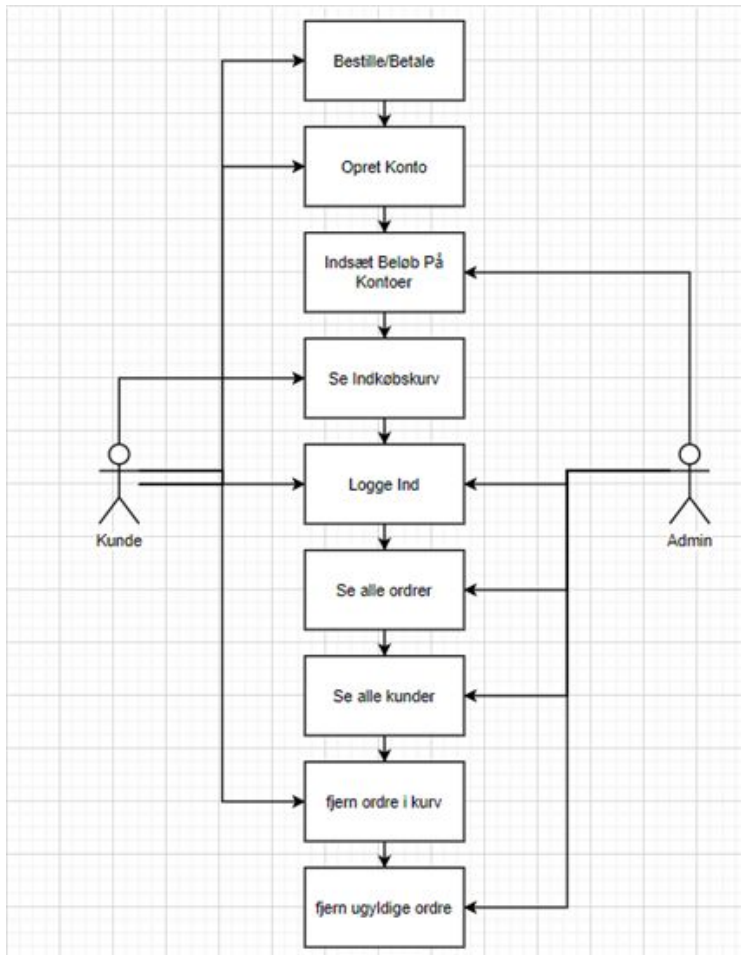
Usercase:

Brugeren og Administratoren har forskellige roller og systemer de har adgang til.

Der vil i navigationsbaren hele tiden være en knap som hedder "Din side", hvor knappen peger hen er bestemt af den rolle som brugeren har, så hvis man ikke har rollen "Employee" tildelt, vil der ikke være mulighed for at tilgå visse administrative funktioner. Skulle man som Customer vælge at trykke på den, bliver man automatisk sendt til `customerpage.jsp`.

Brugerens rolle på hjemmesiden er at oprette en konto så brugeren kan logge ind på et senere tidspunkt og begynde at handle på hjemmesiden ved at vælge og sammensætte sine cupcakes. Brugeren har adgang til indkøbskurven hvor varerne bliver sat ind når kunden bestiller varerne fra shoppen, hvor han derefter kan se en oversigt af sin bestilling og få en samlet pris. Hvis kunden ikke har penge nok på sin brugerkonto, vil der blive informeret om dette, inkl. deres daværende saldo og en henvisning til at få sat penge på den igen for at kunne handle videre. Hvis kunden fortryder en bestilling kan kunden altid gå ind i indkøbskurven og fjerne sin ordrelinje.

Administratorens rolle er at kunne sætte kundens kredit så kunden har mulighed for at handle på hjemmesiden. Derudover har admin en oversigt over aktive ordre der kan admin se alle aktive ordre. Admin kan også gå ind og se de ordre som er betalt, men har også mulighed for at slette en ordre. Inde fra admin kan man se hvilke brugere som er på websitet, derfra kan admin se brugerinformation, som email, password, deres rolle(employee eller customer) og kundens nuværende kredit. Admin holder en oversigt over hvor mange bestillinger der går igennem. Derudover har Admin også mulighed for at fjerne ugyldige ordre fra oversigten, det vil sige ordre som der ikke er gået igennem(ubetalte).

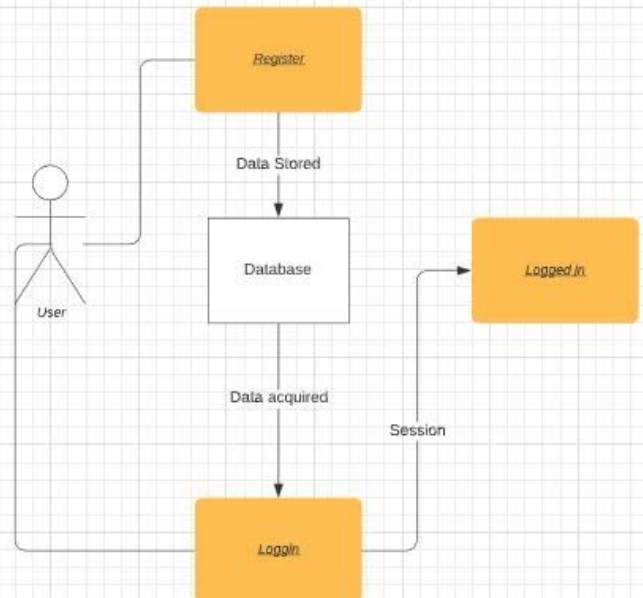


databasen, hvor den forhåbentlig bliver valideret. Hvis hans kombination passer, bliver brugeren logget ind og kan frit navigere på hjemmesiden.

Aktivitetsdiagram:

Aktivitet Diagrammet viser hvordan registrering og login af en bruger foregår. Først registreres brugeren hvor dataen i form af email og password bliver gemt i databasen og brugeren får tildelt et "bruger ID". Brugers kredit er sat til automatisk at være 0, når brugeren bliver oprettet. Efter oprettelse af bruger bliver man automatisk logget ind. Når brugeren igen vælger at besøge siden, skal han logge ind med en kombination af hans email og password, som så bliver sendt til

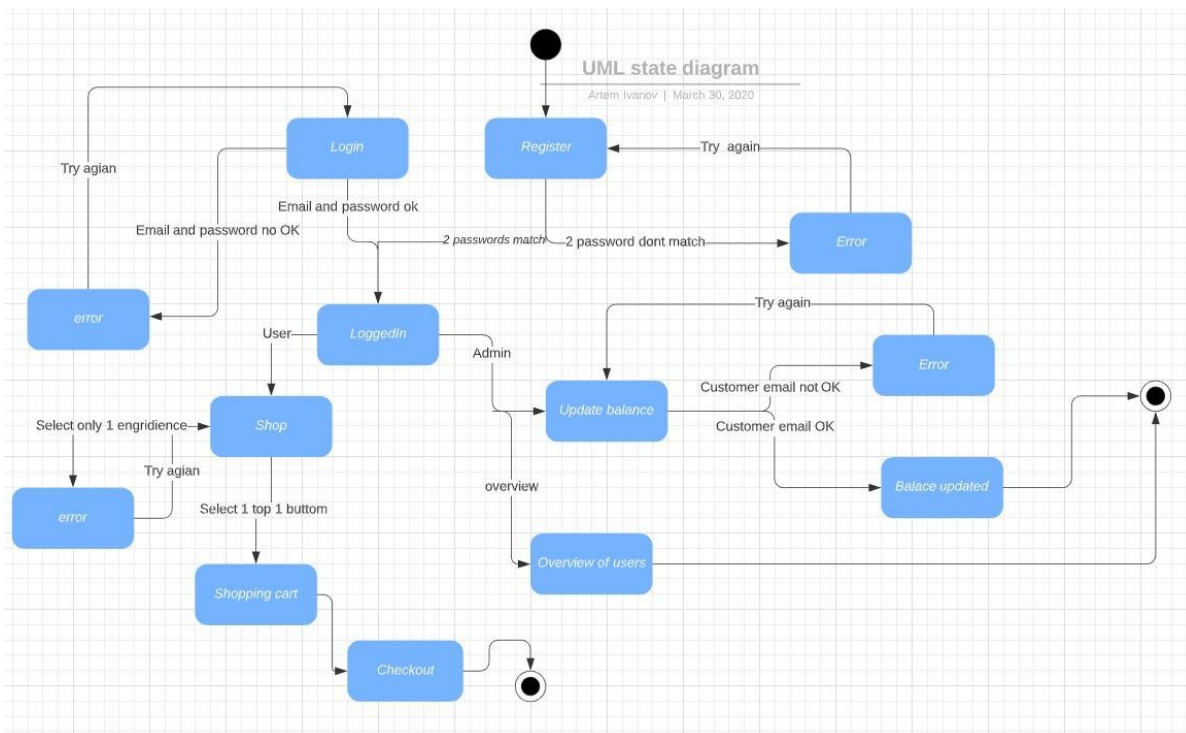
Login/Registrering Aktivitetstestdiagram



State Diagram:

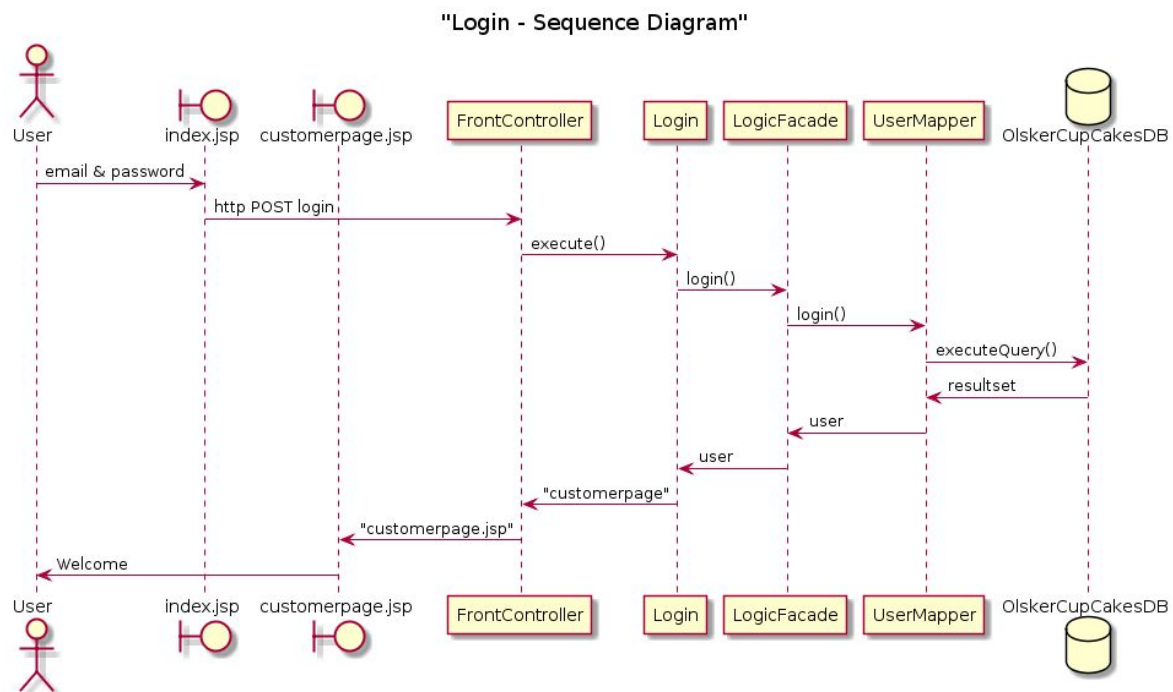
State diagrammet viser hvordan vores program/web application opfører sig når man giver den forskellige input. Som man kan se i starten, har programmet brug for 2 matchende passwords for at kunne gennemføre registreringen af en bruger. Passwords som man taster ind kan enten være ens eller forskellige, hvis passwordene man har tastet ind matcher så bliver man taget til det næste trin af programmet hvor man kan logge ind med sin nye brugerprofil. Dog, hvis passwordene ikke matcher vil man få en fejlbesked om at der kræves to ens passwords for at komme videre og så kan man få lov til at prøve igen.

Sådan fortsætter forløbet med mange af de andre trin som man kan se nedenunder i det vedhæftede diagram.

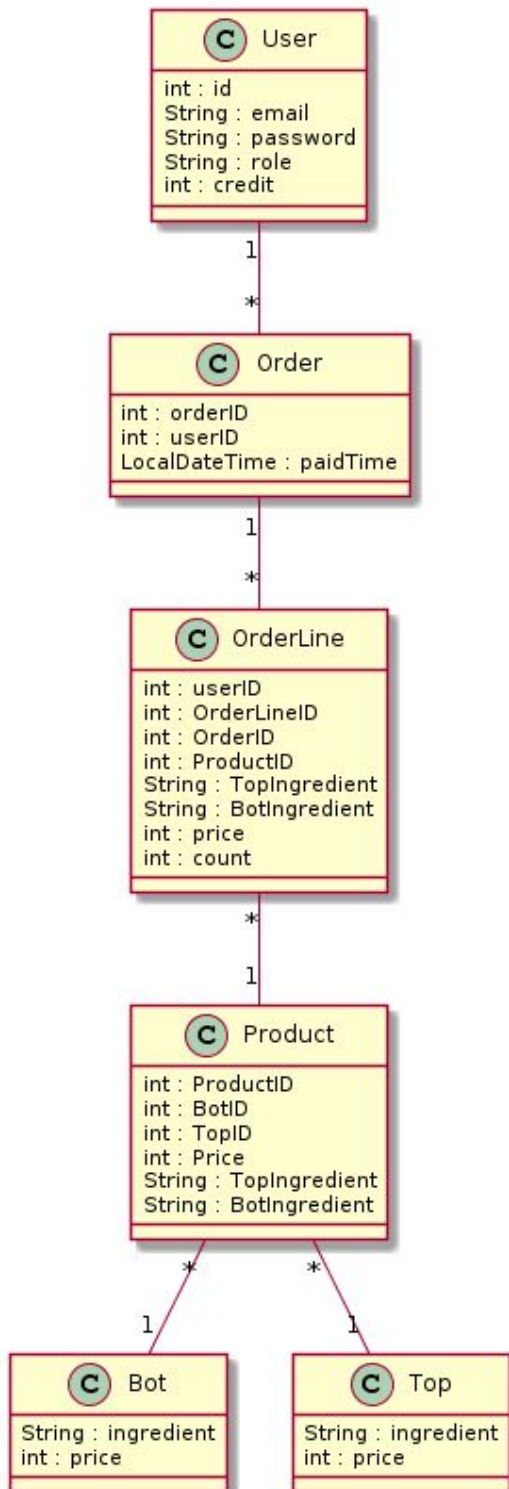


Sekvensdiagram

Sekvens diagrammet viser, hvordan brugerens login oplysninger bliver sendt ned gennem systemets lag til databasen. I databasen bliver login oplysningerne valideret, så den kan hente resten af brugerens oplysninger og sende dem tilbage mod Customerpage.jsp. Man ender dog kun på customer.jsp, hvis brugeren er kunde og ikke administrator. Hvis brugeren er administrator ender man på Employee.jsp. Det bliver valgt om brugeren skal sendes til den ene eller anden forside i Login.



Domæne Model



Særlige forhold

- Hvilke informationer gemmes i session?

Vi bruger hovedsageligt session til at gemme email, så brugeren kan se sin email i hjørnet under hele log in sessionen. Derudover bruger vi session hver gang vi viser en liste af ting. Lister af ting er bare arraylister, som er blevet gemt i sessionen og kørt igennem med et foreach loop i jsp.

Eksempler på sessions fra projekt:

```
"session.setAttribute("email",email);"
```

```
"${sessionScope.email}"
```

```
"session.setAttribute("xy", ordrelinjer);"
```

```
"${sessionScope.xy}"
```

- Hvordan håndterer man exceptions?

Der var en exception der allerede var skrevet i skabelonen, som vi har brugt til al exception handling i projektet. Hvis man kaster denne exception, bliver den samlet op i FrontController, hvor den så sender brugeren til index.jsp med den fejlmeddelelse, som man gav med, da man kodede exception kastet.

- **Hvordan har man valgt at lave validering af brugerinput?**

Validering af brugerinput for registrering, bliver valideret ved at brugeren skal indtaste sin kode 2 gange. Hvis de ikke er ens vil man få en fejlbesked om at koderne skal matche.

Validering af brugerinput for login, bliver valideret ved at bruger skal taste den tilhørende kode som han har forbundet med sit email. Hvis email og koden ikke matcher som de er gemt på databasen vil en fejlbesked blive vist frem, om at enten kode eller email er forkert.

- **Hvordan man har valgt at lave sikkerhed i forbindelse med login?**

Sikkerhed i forbindelse med login foregår ved at koden er censureret under indtastningen, for at folk ikke kan se hvad brugerens kode er mens han taster den ind.

Hvis brugeren har glemt sin kode kan han kontakte en admin for at få sin kode at vide igen. For administratoren har adgang til en log af alle brugere, som er registreret på hjemmesiden.

- **Hvilke brugertyper, der er valgt i databasen og hvordan de er brugt i jdbc?**

Som bruger kan man enten være employee eller customer. I databasen håndterer vi det ved at give brugeren en attribut, der hedder "Role".

Status på implementation

- Vi har fået nået at lave alle de relevante og krævede sider som er obligatoriske til opgaven, men der er visse sider vi selv har tilføjet for at gøre hjemmesiden mere attraktiv som vi ikke har fået nået at kode færdig på. For eksempel "About"
- Vi har ikke nået at få stilet vores side præcis som vi havde forestillet os den skulle se ud.
- Der er stadig nogle errors, som vi ikke har fået handlet ordentligt. Hvis man skriver en mail, der ikke er databasen ind i setCredit fejler sessionen.