Cupcake Rapport

Deltagere:

Navn: Kasper Emil Jeppesen, Email: cph-kj290@cphbusiness.dk, Github: Zieken
Navn: Younes Piskorczyk, Email: cph-yp5@cphbusiness.dk, Github: Younes2630
Navn: Obaydah Elhaj-Moussa, Email: cph-oe134@cphbusiness.dk, Github: obaydahm
Navn: Sinan Jasar, Email: cph-sj381@cphbusiness.dk, Github: sinanj99

Klasse: A

Tidspunkt: 17.03.2019 kl 23:59

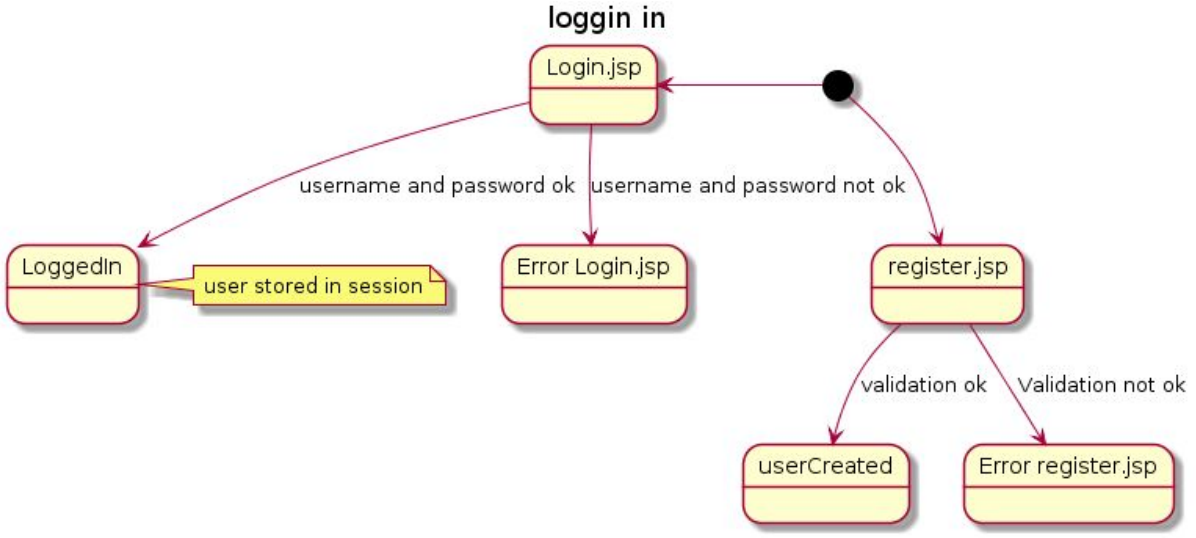
Indledning

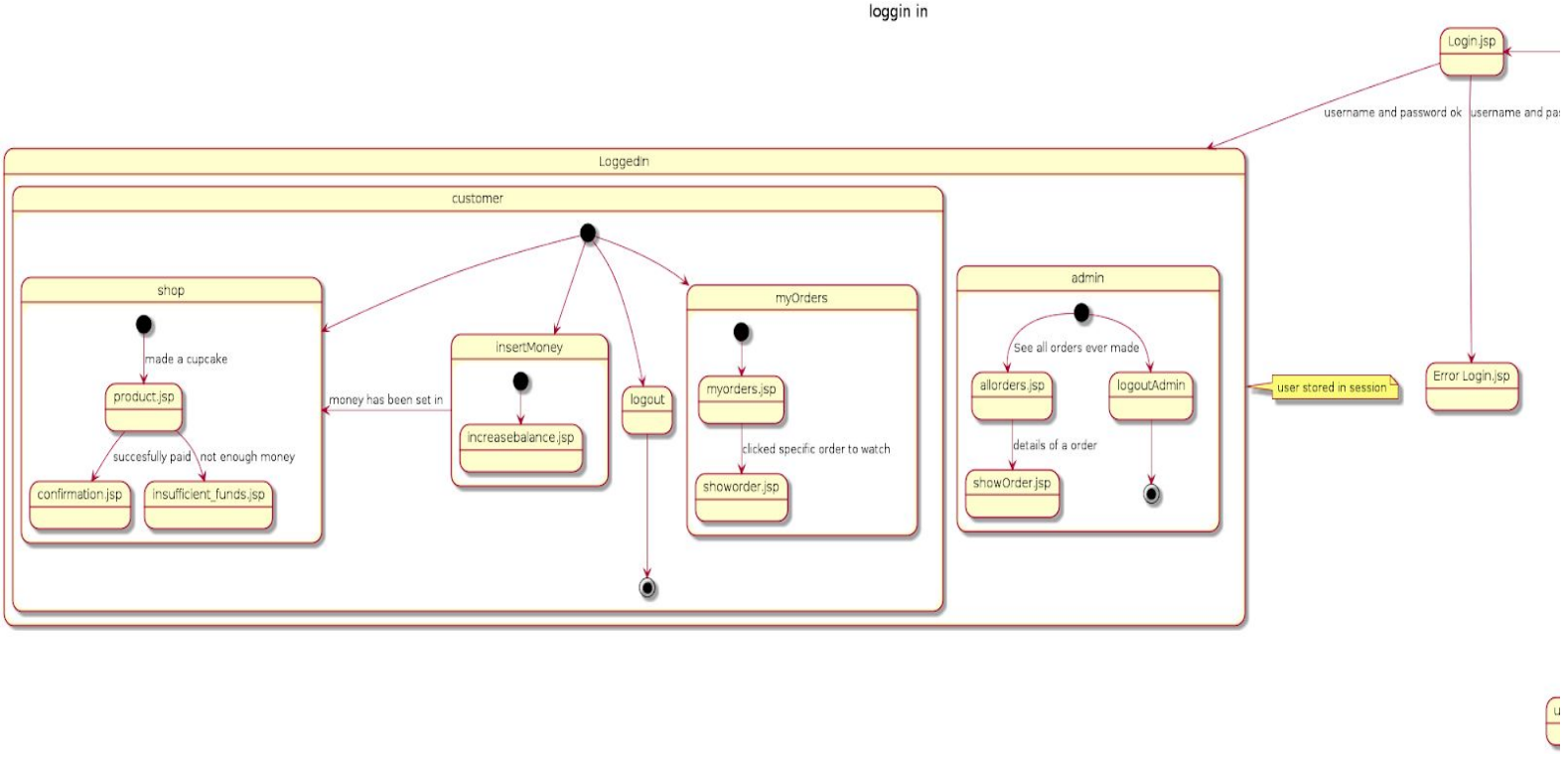
Projektet går ud på at lave en webshop til en virksomhed, der sælger cupcakes. Applikationen fungerer således, at der er kunder, der kan registreres som brugere i webshoppen. Brugeren vælger et brugernavn og password, som vedkommende logger ind med. På brugerens side, har man en saldo, som man skal have penge på for at bestille cupcakes. Man kan løbende tilføje penge til saldoen. Der er en shop-side, hvor man kan tilføje x antal cupcakes, som består af en top og bund, til indkøbskurven. Når ordren er lagt, vil beløbet blive trukket fra saldoen, og bestillingen vil kunne ses i brugerens oversigt over tidligere ordrer. En bruger kan samtidig have administratorrettigheder og kan hermed se en oversigt over alle ordrer.

Vi har benyttet os af Netbeans IDE 8.2, og lavet projektet som en Maven Web Application. Til database-delen har vi brugt MySQL 8.0.15, og JDBC(også 8.0.15) for at forbinde til databasen. Yderligere anvendes Apache Tomcat 8.0.27.0 til at deploye projektet på nettet. Vi har derudover kodet Java og HTML i JSP-filer og brugt CSS til at style programmets brugergrænseflade.

Navigationsdiagram

Tilstandsdiagram

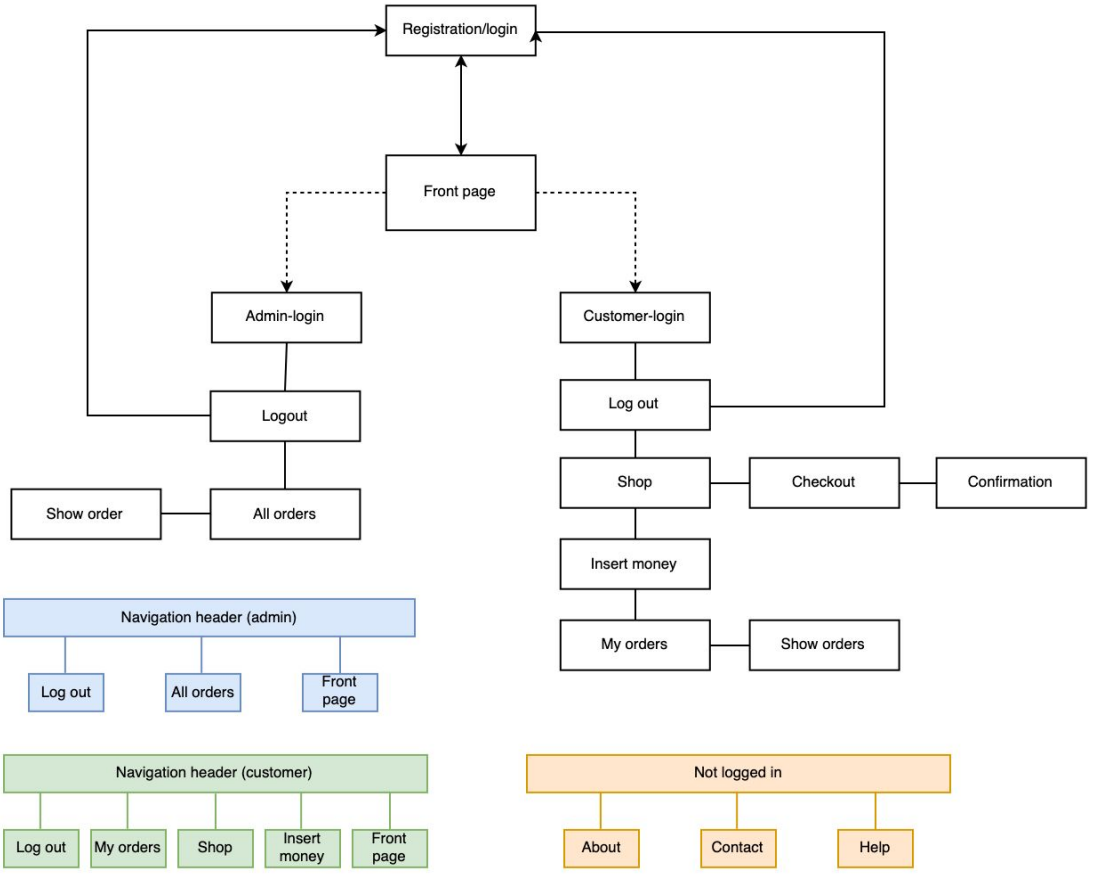




Ovenstående er et tilstandsdiagram, der viser, hvordan man navigerer sig rundt i systemet, og hvilke tilstande man befinder sig i (customer eller

admin), i de forskellige sider. Bemærk at billede 2, blot er en fortsættelse af billede 1.

Oversigtsdiagram



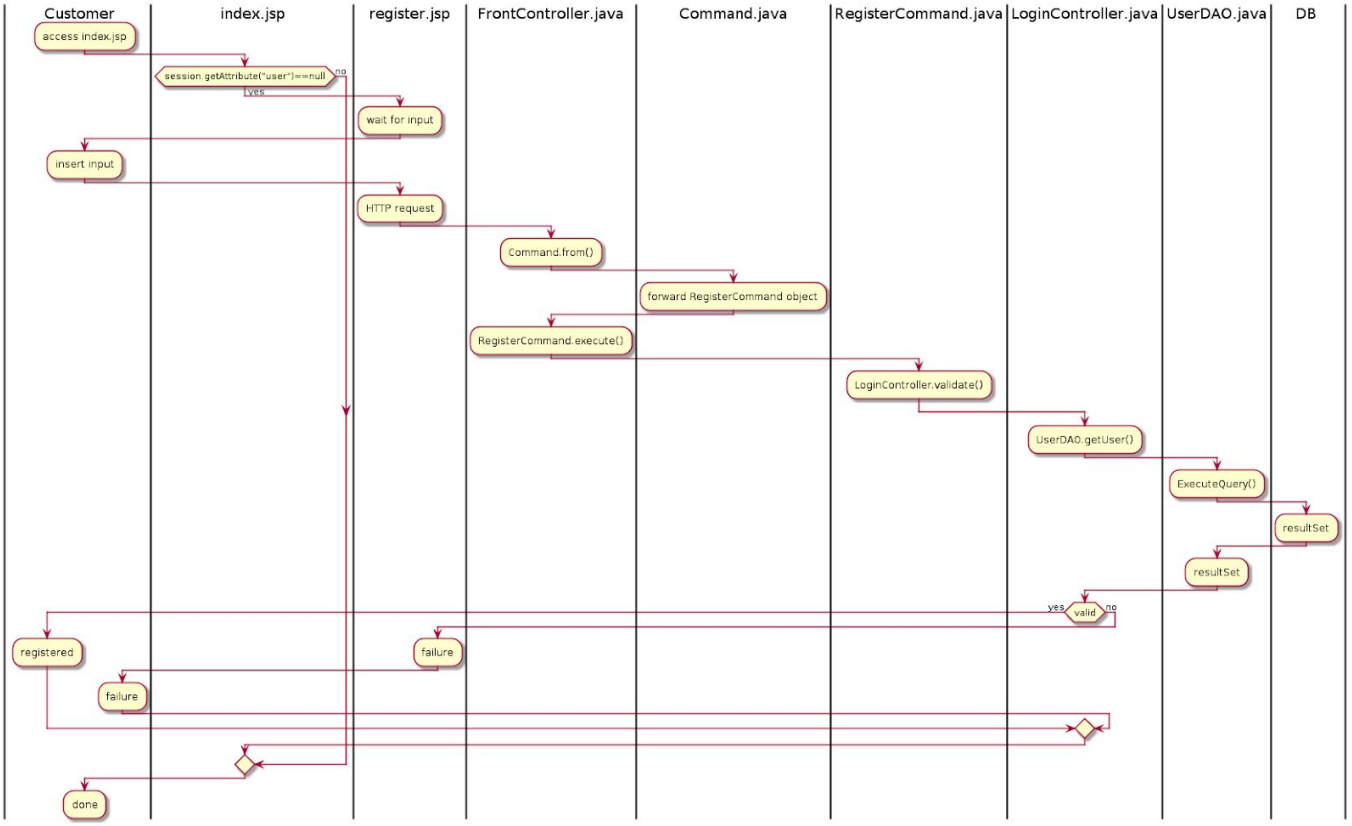
Ovenstående er et mere kundevenligt oversigtsdiagram, som kan give en kunde overblik over systemet.

Som det fremgår af diagrammet er der blevet lavet en navigationsbar, hvis indhold afhænger af, om der er nogen logget ind på session, og hvis ja, hvilken slags bruger der er logget ind på sessionen. Er man ikke logget ind vises den orange, er man logget ind som admin den blå, og er man logget ind som kunde den grønne.

Diagrammet skal læses således, at man logger ind som enten admin eller customer. Herefter bliver de mulige knapper i navigationsbaren listet nedenuder. Ud fra den valgte knap, er der uddybet hvilken process man følger.

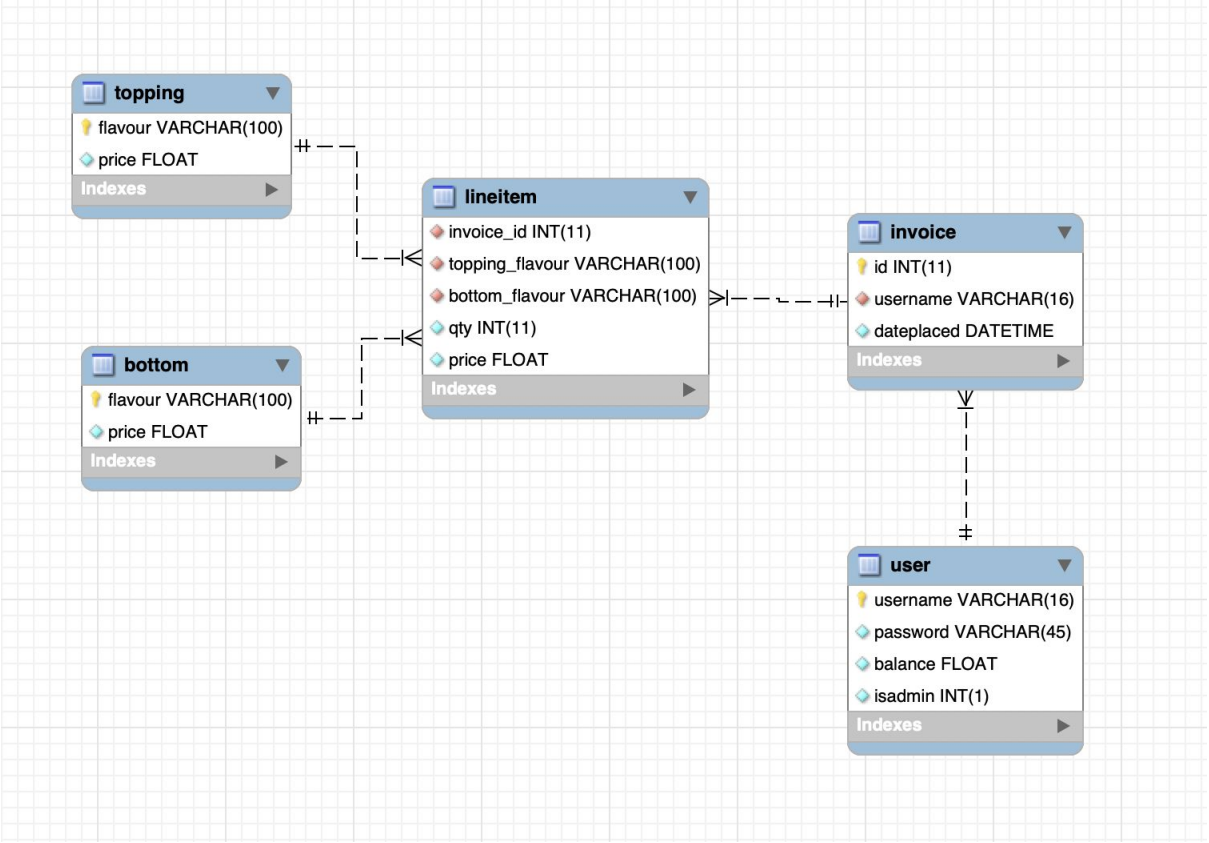
Swimlane

Swimlane - register



Ovenstående swimlane illustrerer de respektive klassers kald af hinanden, når en kunde registrerer sig i systemet. Swimlane for login-processen er udeladt, da princippet er det samme.

ER-Diagram



Ovenstående er et ER-diagram, som viser alle tabeller og relationer.

Primærnøgler

I 'user'-tabellen er attributten 'username' anvendt som primærnøgle, da denne attribut er unik; det samme gælder for 'topping'- og 'bottom'-tabellerne, hvor attributten 'flavour' er anvendt som primærnøgle, da der ikke kan være to cupcakes med samme smag.

Constraints

Alle databasens attributter er angivet som NOT NULL og alt autogenereret id har AUTO INCREMENT slået til.

Constraints på fremmednøgler

- Tabel: invoice - Fremmednøgle: username

  On Update: Cascade: Hvis der implementeres en funktionalitet, der giver mulighed for at ændre en brugers navn eller adgangskode, skal

denne ændring også fremgå i brugerens gamle ordre, da en anden bruger nu kan anvende dette navn (og for at undgå forvirring i det hele taget).

On Delete: No Action: Slettes en bruger fra systemet, skal dataen om brugerens ordrer stadig fremgå i databasen i tilfælde af, at brugeren skulle efterspørge dette.

- Tabel: lineitems - Fremmednøgle: invoice_id

  On Update: Cascade: Opdateres en brugers invoice, skal denne ændring også fremgå i de enkelte lineitems.

  On Delete: Cascade: Slettes en invoice skal dennes lineitems også slettes fra lineitems-tabellen.

- Tabel: lineitems - Fremmednøgle: topping/bottom-flavour

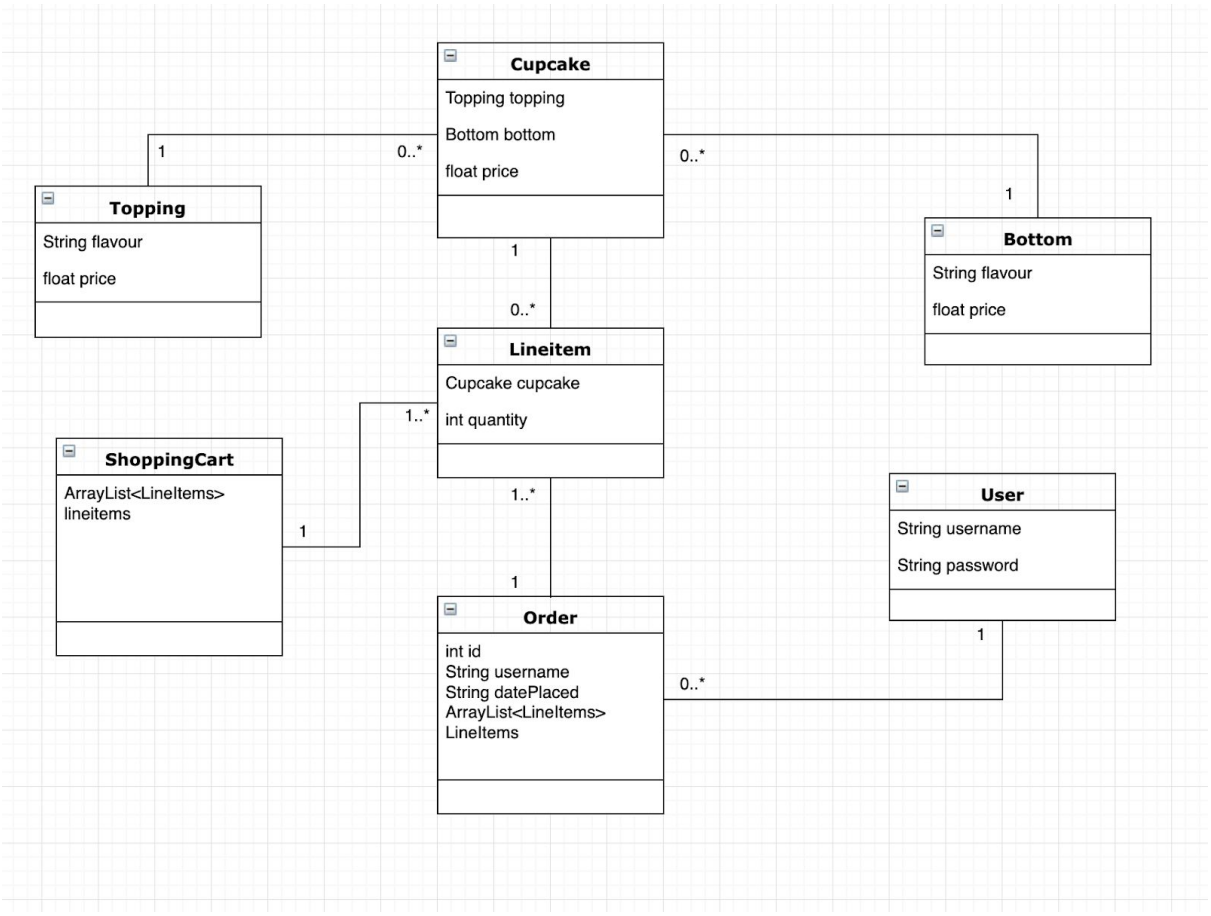
  On Update: No Action: Opdateres en topping eller bottom skal denne ændring ikke fremgå i tidligere ordre.

  On Delete: No Action: Slettes en topping eller bottom skal denne ændring ikke fremgå i tidligere ordre.

Andet

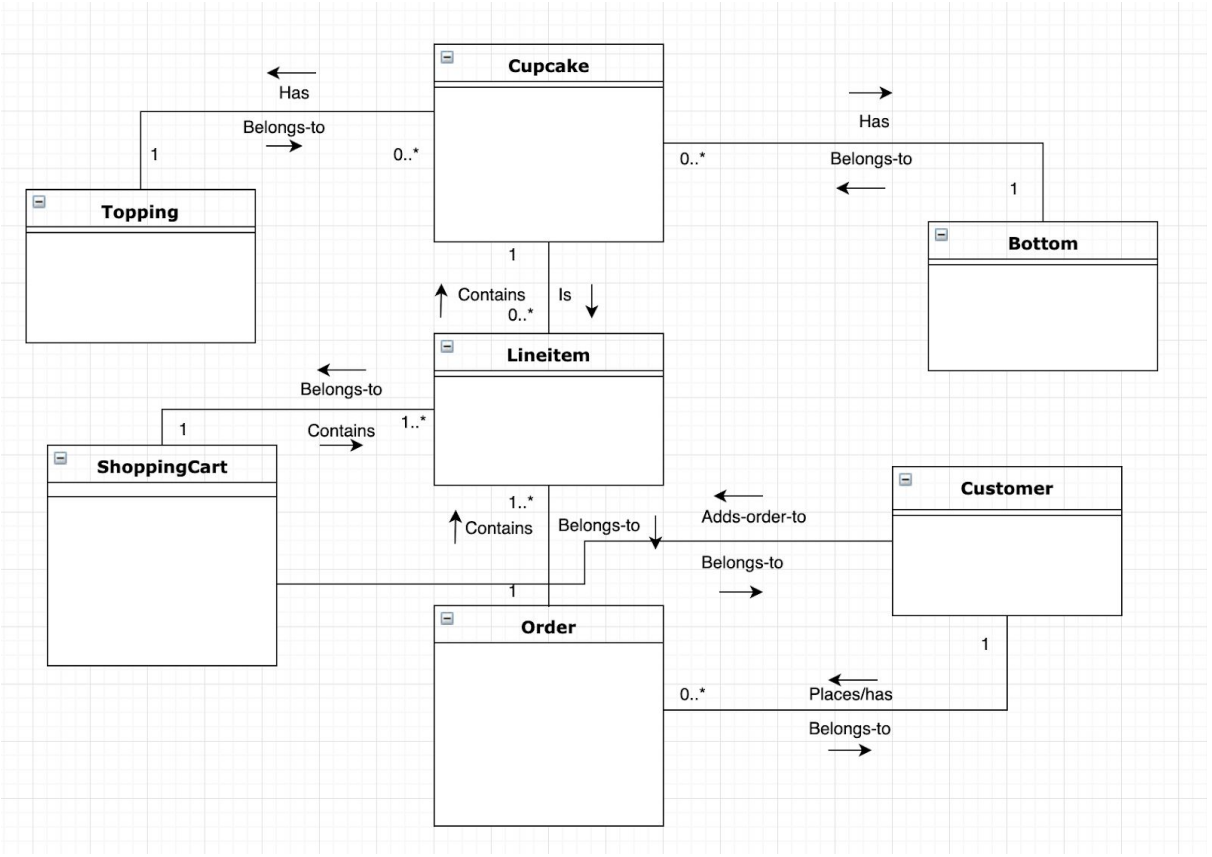
Lineitem-tabellen er ikke i 3. normalform, da der er transitiv afhængighed; attributten 'price' er nemlig afhængig af de non-prime attributter 'qty', 'topping' og 'bottom' og for at sikre systemets integritet vil det være en god idé at rykke price over i en ny tabel og give hver lineitem et autogenereret id. Dette er endnu ikke gjort grundet tidspres.

Klassediagram



Ovenstående er et klassediagram for systemets entitetsklasser.

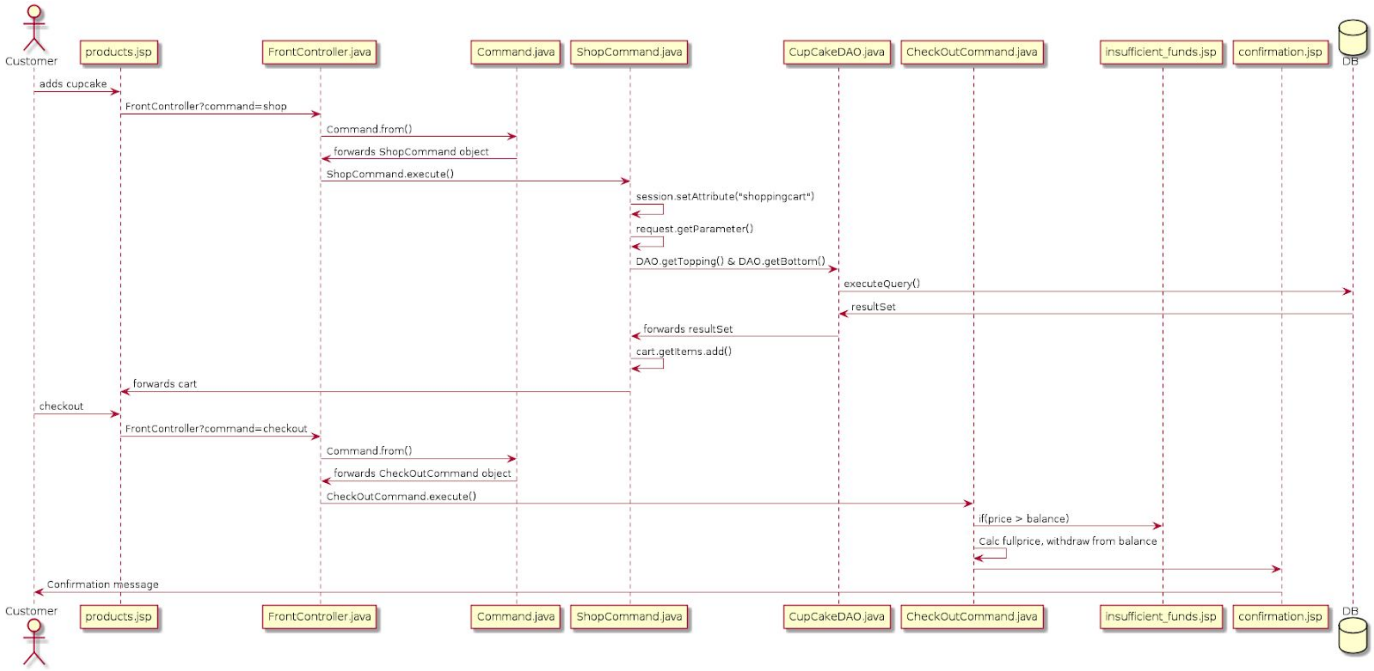
Domænemodel



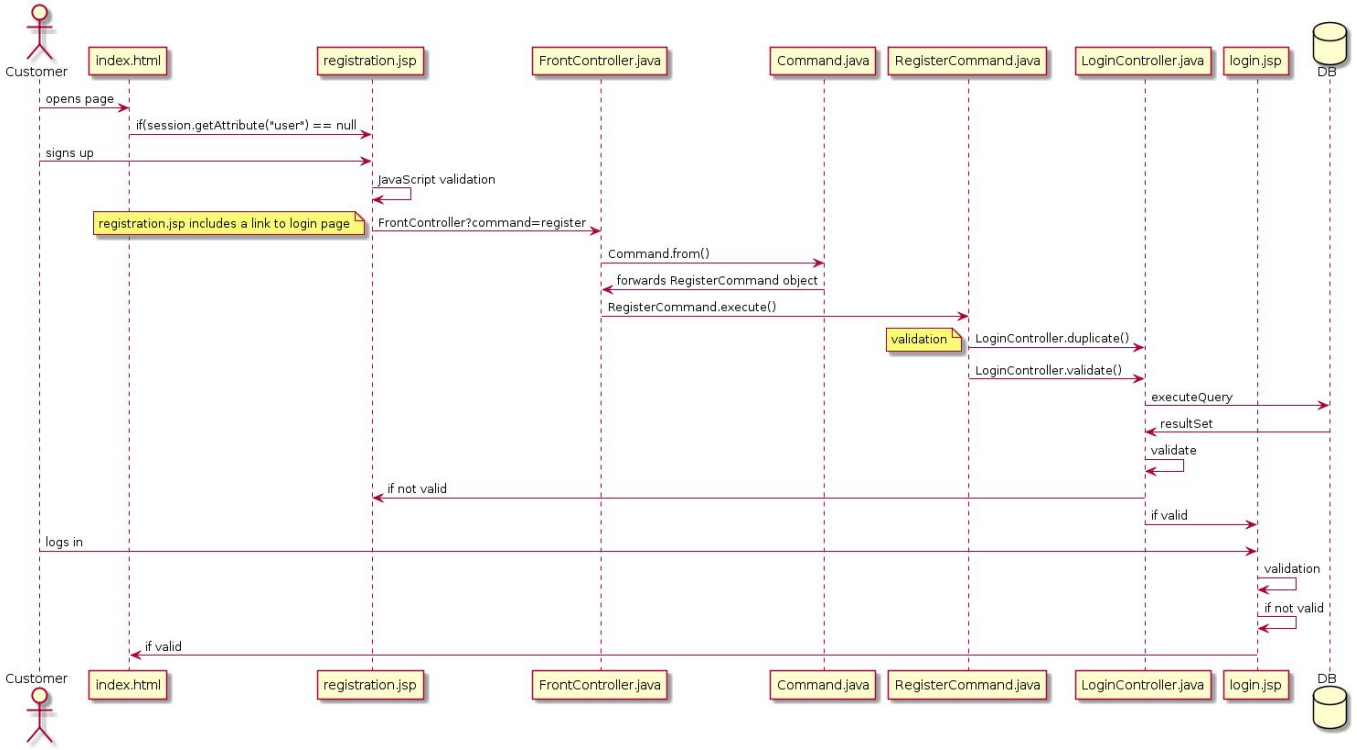
Ovenstående er en domænemodel for systemet.

Sekvensdiagrammer

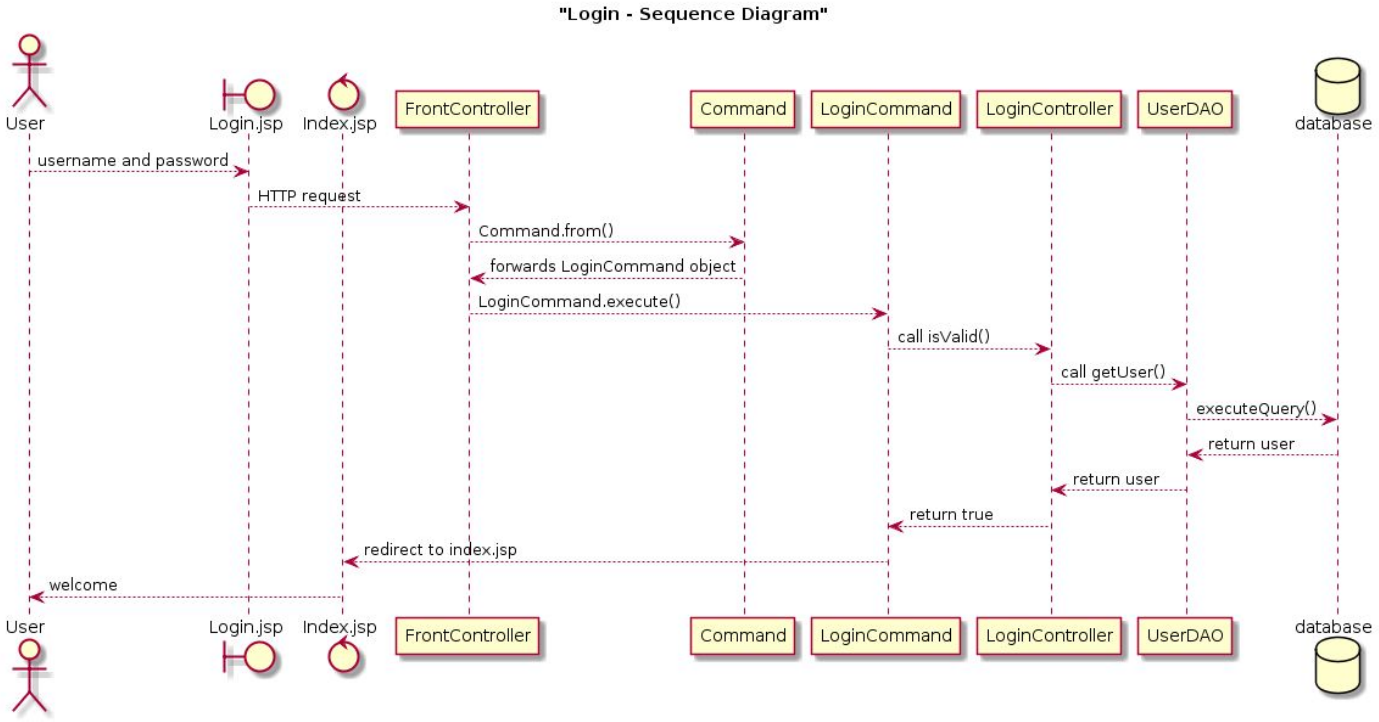
Køb af en cupcake



Adgang til siden - registrering og login



Uddybelse af login-processen



Særlige forhold

Informationer der gemmes i sessions

- User
- Shopping Cart
- Fejlmeddelelser ved validation af login og registrering

Sikkerhed

Validering af bruger-input i forhold til login sker udelukkende på server-siden, da der her blot skal testes for, om en eksisterende bruger matcher inputtet.

Validering af bruger-input i forhold til registration af en ny bruger sker som udgangspunkt på client-siden i register.jsp med javascript, men som en ekstra sikkerhed er der også blevet lavet validering på server-siden med Java; der er blevet lavet 2 klasser i logik-laget, hhv. Username- og PasswordChecker, med metoden *boolean validate(String username)*, der matcher et brugernavn med et REGEX.

```
    public class UsernameChecker {
    public static final Pattern VALID_USERNAME_REGEX =
        Pattern.compile("^[a-z0-9_-]{3,15}$", Pattern.CASE_INSENSITIVE);

    public static boolean validate(String username) {
        Matcher matcher = VALID_USERNAME_REGEX .matcher(username);
        return matcher.find();
}
}
```

Yderligere er der i logik-laget defineret en metode, som checker om et givent brugernavn allerede eksisterer i databasen.

```
public static boolean duplicate(String username) {
    return UserDAO.getUser(username).getUsername().equals(username);
}
```

Slutteligt er der lavet en metode, der tjekker om brugerens brugernavn stemmer overens med brugerens password.

```
public static boolean isValid(String username, String password) {
    if (username == null || username.isEmpty()) {
        return false;
    }
    if (password == null || password.isEmpty()) {
        return false;
    }
    User user = new UserDAO().getUser(username);
    return password.equals(user.getPassword());
}
```

Brugertyper

Attributten isAdmin i databasens user-tabel fortæller os om en bruger er admin eller ej. Attributten kan have værdien 1 eller 0 (1 = Ja && 0 = Nej). Er man admin, har man rettigheder til at se andre brugeres ordre.

Status på implementation

- Styling

  JSP siderne allorders, showorders, myorders, insertmoney og index.html mangler styling. Endvidere mangles brugergrænseflade og backend til sektionerne 'About', 'Contact' og 'Help'.

- Javascript

  Der mangler JavaScript, som, ved registrering, tjekker om brugernavnet og/eller adgangskoden indeholder ugyldige tegn, samt JavaScript til validation af login. Yderligere skal der oprettes et array af ordre på admin-siden og laves 2 knapper, der kan sortere ordrerne på dato og navn. Slutteligt mangler der en knap, som fjerner cupcakes fra kurven, og et input-felt ud for hver lineitem i kurven, hvor man kan ændre antallet af cupcakes.

- CRUD

  Enkelte CRUD metoder mangler til de forskellige klasser.

  Tabeller:

  User: Kunder mangler muligheden for at kunne ændre sin adgangskode, og evt. slette sin bruger.

  Invoice + lineitems: Kunden kan lægge ordrer, tilgå og se dem, men ikke redigere på dem eller slette dem.

  For at spare data er attributterne 'username' i user-tabellen og 'flavour' i hhv. topping- og bottom-tabellen blevet lavet som primærnøgler - dette skal dog ændres, hvis der senere implementeres en funktion, der tillader brugere at redigere sit navn og admins at redigere på cupcakes. Gøres dette skal der laves et autogenereret id i databasen, der i stedet skal fungere som primærnøgle.

- Andet

  En planlagt ændring er at fjerne linket til login.jsp på registration.jsp-siden og i stedet lave et link fra login.jsp til registration.jsp