

Studypoint opgave:

Olsker Cupcakes

Deltagere:

Magdalena Aleksandra Wawrzak

cph-mw216@cphbusiness.dk,

GitHub: magda(som branch), kasarama(som brugernavn)

Mona Jakob Meshal(delvis)

cph-mm551@cphbusiness.dk,

GitHub: mona, CupCake(som branch)

Klasse D

Udleveret: 12.03.2020

Arbejdstid: til og med 30.03.2020

Indledning

Formålet med projektet er at studerende bruger den hidtil erhvervede viden samt opnåede færdigheder, inden for frontend og backend programmering, databaser, operative systemer, serverer, servlets, droplets, HTML mm.

Opgave er blevet udleveret i form af beskrivelsen af en virksomhed og dets behov for et IT system og løsningen skal udvikles med udgangspunkt i "Kaspers Command-Pattern skabelon"

Data skal gemmes i en database, som skal normaliseres på 3. normalform. Kildekoden deles på GitHub. Det færdige produkt skal udvikles i Java, MySQL, HTML, CSS, Twitter Bootstrap og køre på en Tomcat webcontainer.

Det færdige produkt skal i sidste ende køre på en Droplet hos Digital Ocean

Opgaven løber fra og med d. 12.03.2020 til og med, deadline den 30.03.2020 kl. 23:55

Udover de tekniske opgaver, praktiserer de studerende gruppearbejde og samarbejde online ved brug af gitHub, Zoom, eller lignende.

Baggrund

Olsker Cupcakes er et bageri på Bornholm, virksomheden drives efter økologiske principper.

Konceptet er bundet op på smag og valgfrihed, forstået på den måde at kunderne selv sammensætter kage og topping, udfra et sortiment med forskellige varianter, prisen varierer og bestemmes udfra pris på henholdsvis bund og topping.

Kundernes krav til systemet:

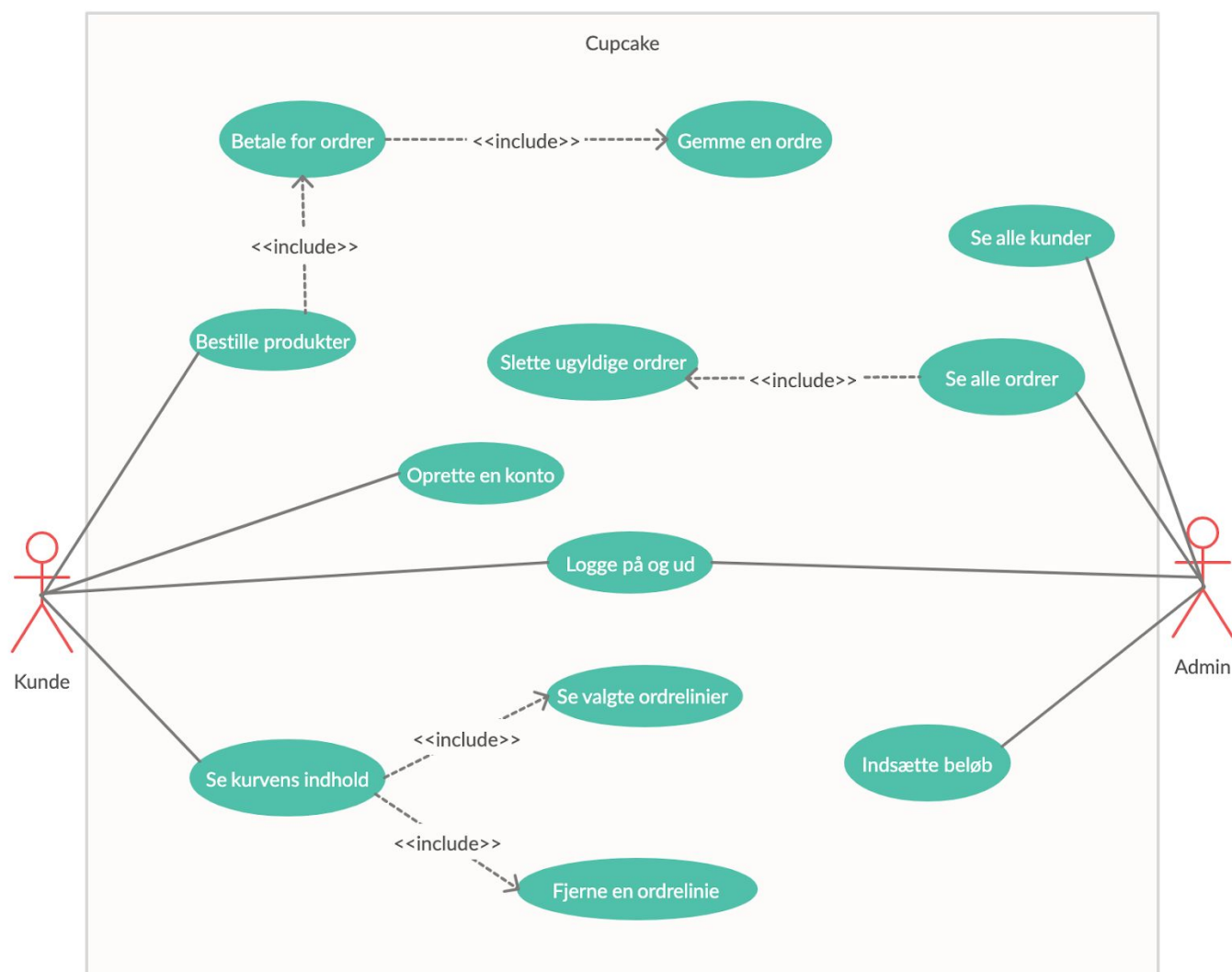
Bageriets kunder skal kunne:

- Bestille og betale en eller flere cupcakes med valgfri bund og top
- Oprette en konto/profil - nødvendigt for at kunne betale og gemme en ordre
- Lægge vare (cupcakekes) i en indkøbskurv, med antal og samlet pris på hele ordren
- Justere en ordre ved at kunne fjerne enkelte vare fra indkøbskurven
- Logge på systemet med e-mail og kodeord
- Se e-mail/logg in oplysninger på hver side i logg ind perioden
- Få indsat beløb på sin konto igennem en administrator

Administrator skal kunne:

- Logge på systemet med e-mail og kodeord
- Se sin e-mail på hver side når de er logget ind
- Indsætte beløb på en kundes konto
- Se alle ordrer detaljeret i systemet
- Se alle kunder samt deres ordrer i systemet
- Fjerne en ordre med forskellige årsagskoder

Billede 1. Use Case Diagram



Teknologi valg

Java 8

IntelliJ IDEA Educational Edition 2019.3.1

Maven build automation tool

Tomcat 8.5.51 local server configuration

MySQL 8.0 Command Line Client

MySQL Workbench 8.0 CE

DigitalOceans Droplet:

Ubuntu 18.04.3 LTS

MySQL Ver 14.14 Distrib 5.7.29, for Linux (x86_64) using
EditLine wrapper

Server version: 7.29-0ubuntu0.18.04.1

Apache Tomcat Version 9.0.30

FilleZilla 3.47.2.1

Krav

Systemet skal gøre det muligt for kunderne

- At kunne vælge de forskellige komponenter som produktet består af
- At kunne oprette sig som en ny bruger
- At kunne logge på systemet
- At have saldo på sin konto
- At bestille produkter og betale for ordrer online

Systemet skal gøre det muligt for administratoren

- At logge på systemet
- At se alle kunde profiler
- At se alle ordrer
- At kunne slette en ordre, med forskellige årsagsangivelser
- At kunne opdatere kundesaldoer

Systemet vil gemme kunder og deres ordrer, dette vil garantere datalagring og mulighed for data manipulation.

Dette vil imødekomme virksomhedens grundlæggende behov inden for online kundeservice, og vil tilmed åbne muligheder for et bredt fremtidigt samarbejde, samt videreudvikling og eventuel udvidelse af systemet og dets funktionalitet.

Scrum user stories(i samarbejde med Mona Jakob Meshal):

ID	1	Name	Log in	Priority	High
Primary Actor	Customer/Administrator				
Pre-condition	The user should have a valid account to log into the system. System must be connected to DB				
Post-condition	The system should log the user in and show the login page.				
Description	To log into the system, the user should have a valid username and password. The user will click on the "login" button from the start page, or will be redirected from cartpage. The system will then display a "login" page with fields for e-mail address and password. The user will type the login information and click on the "log in" button. The system will evaluate the input and display the home page (if request came from homepage) or cartpage(if request came from cartpage) if it's correct or if incorrect, go back to the "login" and display an error message.				
Scenario Steps	Actor			System	
	1. Click on "Log ind" button from the header/ Click on "Bestil" button from the cartpage			2. Display the "log in" page.	
	3. Type in the login information.			4. Evaluates the typed login information.	
				5. Displays home or cart page if input is correct or display an error msg on the login page.	

ID	2	Name	Create a new user account	Priority	High
Primary Actor	Customer				
Pre-condition	The system should display a login page that includes a "login" and "Registrer" button. System must be connected to DB				
Post-condition	The system must create a new user account.				
Description	The user will click on the "Registrer" button from the start page. The system will display the register page. The user will then type an email address and a password twice as well as first and last name. The system will evaluate the inserted information by accessing the database. An error message will be returned if the email is in use or if the passwords don't match. Else, the system will create the account and display the menu page.				
Scenario Steps	Actor			System	
	1. Click on "Registrer" button from the start page.			2. Display the register page	
	3. Type in the information.			4. Evaluate the typed information	
				5. Display an error message on register page or create a new user account and display customer page	

ID	6	Name	See orders	Priority	Medium
Primary Actor		Admin			
Pre-condition		User must be logged in with a valid account. System must be connected to DB			
Post-condition					
Description					
Scenario Steps		Actor		System	
		1. Click on "Vis ordrer" on the header or "Vis mig dem alle" from admin page		2. Display ordersAdmin page with order details from DB	

ID	7	Name	Delete order	Priority	Low
Primary Actor		Admin			
Pre-condition		User must be logged in with a valid account. System must be connected to DB			
Post-condition		Order must be deleted from database			
Description		While user is on the ordersAdmin page types the text in a field and clicks on the "Slet" button at the chosen order. System displays msg with confirmation and returns to ordersAdmin page			
Scenario Steps		Actor		System	
		1. On ordersAdmin page type a comment text and click on the "Slet" button at the chosen order		2. Display msg with confirmation and returns to ordersAdmin page	

ID	8	Name	See customers	Priority	Low
Primary Actor		Admin			
Pre-condition		User must be logged in with a valid account. System must be connected to DB			
Post-condition		System must show the list of customers			
Description		User clicks on "Vis kunder" button on to see the list of customers with their data			
Scenario Steps		Actor		System	
		1. Click on the "Vis kunder" button on header or "Jeg vil gerne kigge!" on "admin" page		2. Display "allCustomers" page with customer details from DB and the	

Domæne model og ER diagram

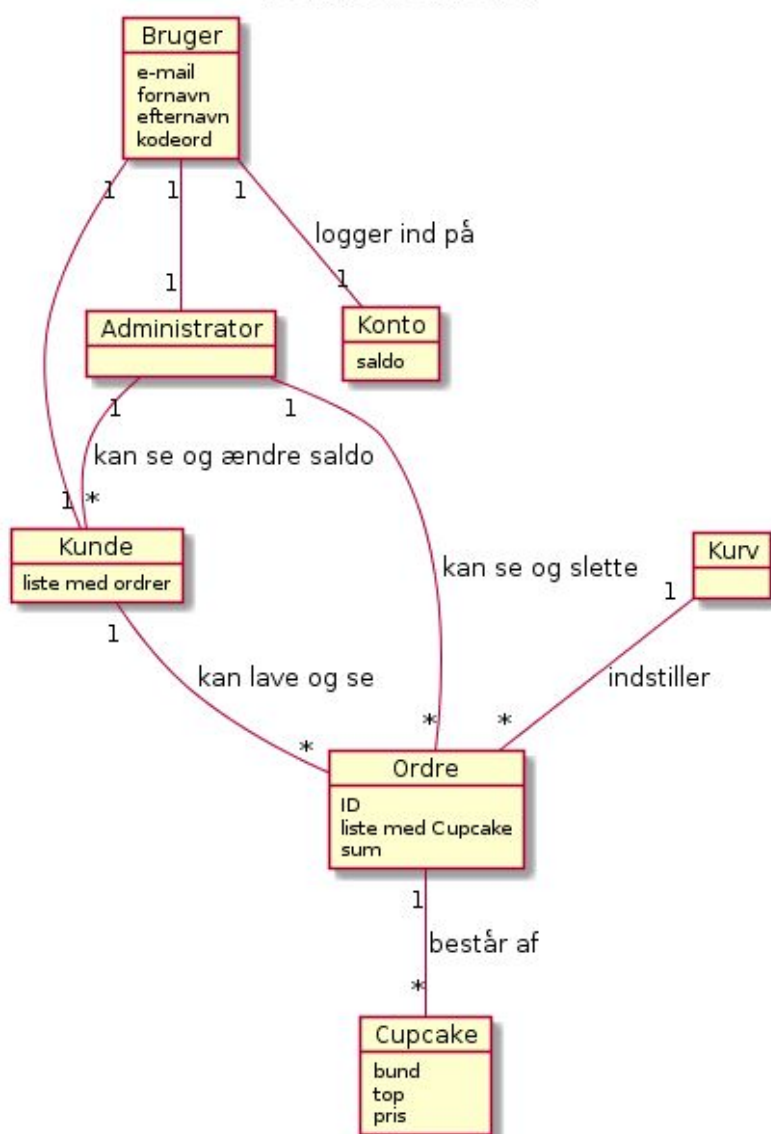
Domæne Model:

Domænemodel Diagram

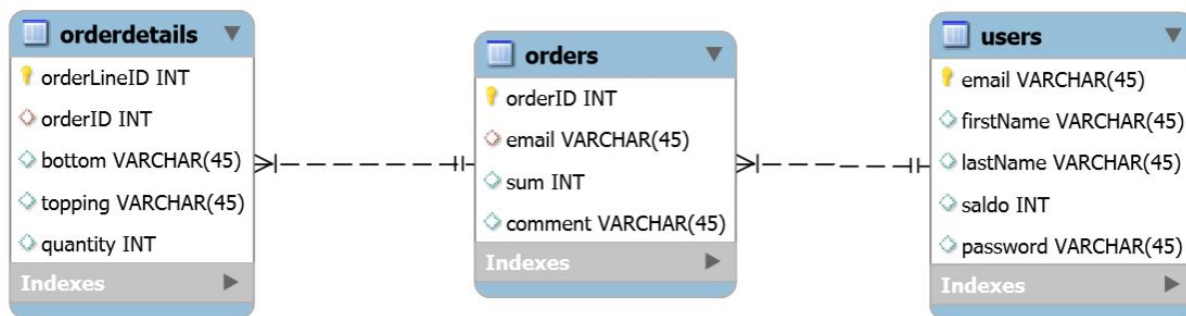
Domæne Model

Beskrivelse af relationer mellem objekterne i modellen:

- En Bruger beskrives med e-mail adresse, fornavn og efternavn. Den kan logge ind til systemet med e-mail og kodeord.
- Bruger kan enten være en Administrator eller en Kunde.
- En Kunde har en konto med saldo på den.
- En Konto tilhører en Kunde.
- En Kunde kan lave og se flere Ordre.
- En Ordre kan lave af en Kunde.
- En Ordre beskrives med et ID nummer, og den har en liste af valgte Cupcakes samt sum af priser af alle vare på listen
- En Ordre indstilles gennem en Kurv. En Kurv kan indstille flere Ordre.
- En Ordre består af flere Cupcakes.
- En bestilt Cupcake kan gemmes på en liste af en Ordre.
- Cupcake består af bund, top og pris.
- En Administrator kan se og slette flere Ordre. Ordre kan administreres af en Administrator.
- En Administrator kan se flere Kunder. Kunder kan ses af en Administrator.



ER Diagram cupcake database



Databasen opfylder kravene af 3. normalform. I users tabellen bruges e-mail som primær nøgle. Systemet kræver at e-mailadressen er unik for hver bruger og derfor kunne der udelades en ekstra kolonne. Ordre og ordrerdetaljer bruger numerisk, automatisk genereret ID som primær nøgle (orderID og orderdetailsID).

Tabeller har en til mange relationer: En email kan være en fremmednøgle i flere rækker i ordre tabellen, et orderID kan være en fremmednøgle i flere rækker i orderdetails tabellen.

De brugte constraints er INT* til de numeriske værdier. Prisen på komponenterne er angivet som heltal med mulighed, for at indsætte beløbet på en konto, saldo er derfor begrænset til heltal. Heltal lagt sammen giver altid heltal og derfor er summen af ordren er også et heltal, primærnøgler, antal af cupcakes skal være heltal.

De ikke numeriske data gemmes som VARCHAR(45) Længden er default defineret af MySQL Workbench. Der var ingen krav til hvor lange data en user må angive.

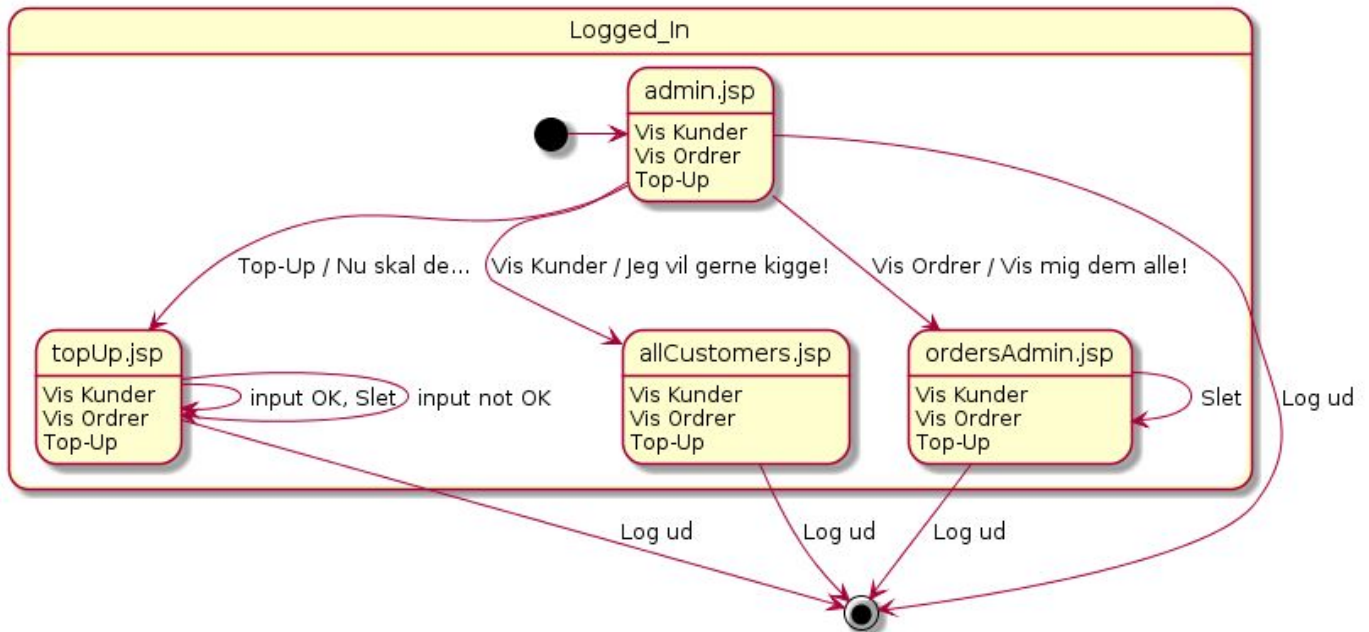
*Der var lavet en fejl ved programmeringen, hvor der blev brugt Integer som type af attribute af en Cupcake object. Priser burde være gemt som attribute af typen double. Fejlen blev håndteret ved at give begrænset mulighed for input for brugeren og har ingen betydning for funktionaliteten af systemet, dette ville dog kunne dog skabe problemer for videreudviklingen af systemet.

Til den specifikke opgave er løsningen dog tilstrækkelig.

Fejlen er ikke rettet pga. Manglende arbejdskraft.

- Programmet starter på index.jsp siden. Der kan bruger vælge at logge ind, tilføje nogle Cupcakes til indkøbskurven eller at se siden af indkøbskurv.
- Trykkes der på "Log ind" knappen, bliver man sendt til login siden. Der skal man indtaste sin e-mail og sit kodeord for at logge ind, eller trykke på "Registrer knap" for at registrere sig som en ny kunde/bruger. Hvis de indtastede data er korrekte, bliver man sendt tilbage til hjemmesiden
- Brugeren kan vælge sin vare ved at vælge bund, top og antal og derefter trykke på "Læg i kurv" knappen. Cupcake/-s bliver derefter tilføjet til indkøbskurven
- Ved at trykke på et billede med indkøbsvogn bliver man sendt videre til indkøbskurven
- Hvis kurven er tomt vil man se en knap med teksten "Find nogle Cupcakes her". Den sender brugeren tilbage til hjemmesiden.
- Hvis der er Cupcakes i kurven, kan man se de valgte Cupcakes og slette en slags ad gangen ved at trykke på "Slet" knappen, herefter vises det opdaterede indhold af kurven.
- Ved at trykke på "Tilføj flere" knappen, bliver man sendt til hjemmesiden/startside.
- Hvis brugeren ikke har været logget ind i forvejen, sendes brugeren videre til loginsiden når der trykkes på "Bestil". Efter logge in viser systemet indkøbskurv siden igen.
- Hvis en bruger er logget ind og bruger knappen "Bestil" valideres brugerens saldo og vises på indkøbskurv siden, hvis saldo er for lav eller på bekræftelses siden hvis der er nok penge på brugerens konto
- På confirm.jsp er der to valgmuligheder: for at vende tilbage til indkøbskurv trykkes "Tilbage til indkøbskurv", hvor ordre kan justeres, for at bekræfte ordre trykkes på "Betalt".
- Ved at klikke på "Log ud" knappen på alle sider, bliver data fra sessionen slettet og systemet viser startside
- På de sider hvor brugeren er logget ind, vises der en fælles navigations bar. Den bliver beskrevet på de næste diagrammer.

Navigation Diagram 2.
Navigation from admin page

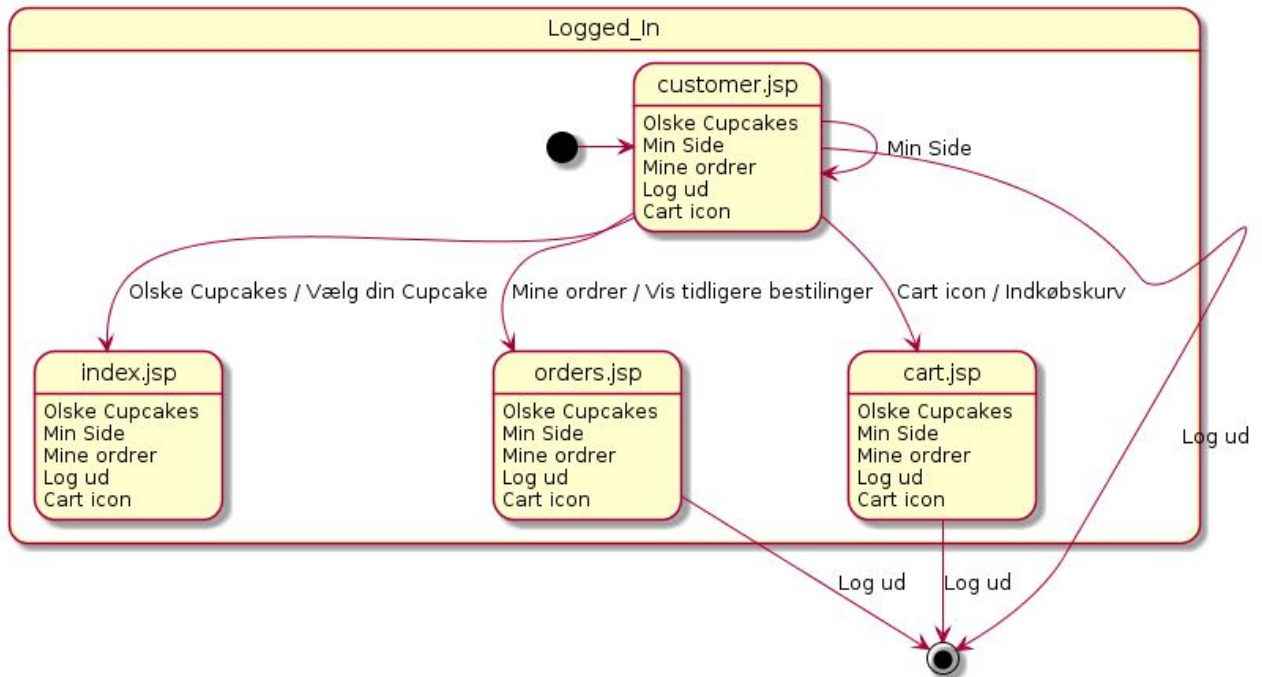


Alle sider på diagrammet er tilgængelige kun for bruger logget ind som admin@olsker.cupcakes

- Hver side bruger fælles navigation bar med knapper: Vis Kunder, Vis Ordre, Top-Up, Log ud*. Hver knap sender til sin side, som vist ved admin.jsp. Bruger kan bevæge sig rundt på systemet på de sider, admin har adgang til.
- Når der klikkes på "Vis Kunder" eller "Jeg vil gerne kigge!" på admin siden, vises der oversigt over alle kunder i systemet.
- Når der klikkes på "Vis Ordre" eller "Vis mig dem alle!" på admin siden, vises der oversigt over alle ordre.
- Hver enkelt ordre på ordre oversigt siden har knappen "Slet". Bruger kan indtaste tekst og trykke på "Slet". Systemet viser herefter opdateret oversigt over ordre
- Når der klikkes på "Top-Up" eller "Nu skal de (..)" på admin siden, vises siden til at indsætte valuta på en brugerkonto. Bruger taster e-mail og vælger beløb. Hvis e-mail er korrekt, vises brugerens nye saldo, ved brug af forkert emailadresse vises en besked om ugyldig e-mail adresse

*Log ud" knappen mangler på diagrammet.

Navigation Diagram 3.
Navigation from customer page

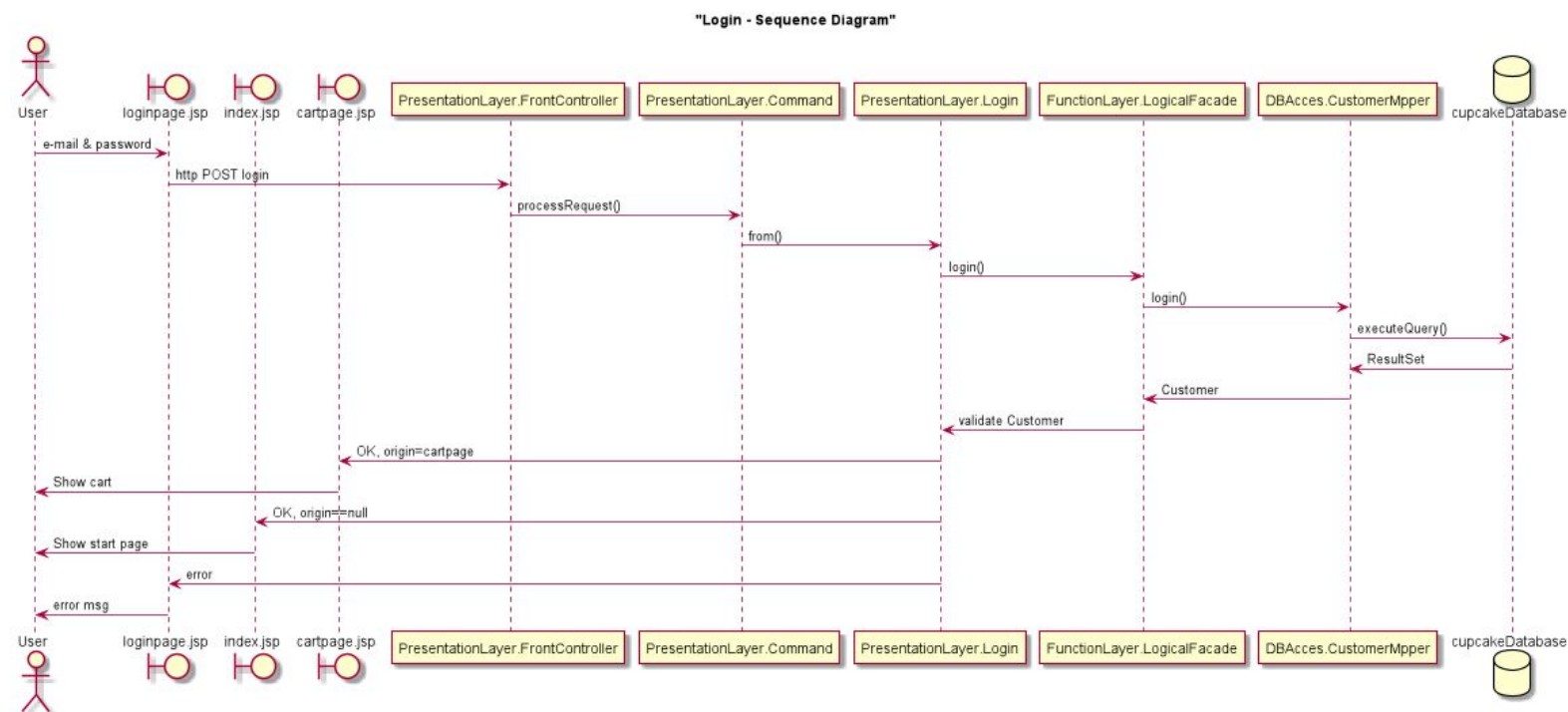


Alle sider på diagrammet er tilgængelige kun for bruger logget på systemet ekskl. `admin@olsker.cupcakes`

- Hver side bruger fælles navigation bar med knapperne: "Olsker Cupcakes", "Min Side", "Mine ordrer", "Log ud" og indkøbsvogn billede (beskrevet ved Navigation Diagram 1). Hver knap sender til sin side, som vist ved `customer.jsp`. En bruger der er Logget ind kan bevæge sig rundt på systemet på sider nævnt på diagrammet.
-
- Når brugeren klikker på "Olske Cupcakes" eller "Vælg din Cupcake" på kundesiden, vises hjemmesiden/startsidens.
- Når brugeren klikker på "Min side" vises kundesiden
- Når brugeren klikker på "Mine ordrer" eller "Vis tidligere bestillinger" på kundesiden, vises en oversigt over kundens ordrer.

Sekvens diagrammer

Sekvens Diagram 1



- Log ind processen begynder på loginpage.jsp -bruger taster e-mail og kodeord og trykker "Log ind" knap - HttpRequest bliver sendt med http protokole til PresentationLayer.FrontController
- I FrontController kaldes på PresentationLayer.Command med processRequest()
- I Command udføres from() og sendes til PresentationLayer.Login
- Login læser e-mail og kodeord fra HttpRequest og bruger dem i login(), som kalder på FunctionLayer.LogicalFacade
- LogicFacade bruger e-mail og kodeord for at kalde på DBAccess.CustomerMapper
- CustomerMapper udfører sql query med e-mail og kodeord til cupcake database
- Databasen svarer med ResultSet til CustomerMapper.
- CustomerMapper instantierer Customer objekt ud fra ResultSet og sender den til LogicalFacade
- LogicalFacade sender objektet videre til Login Servlet.
- På Servlet validerer modtaget data.
 - Hvis Customer er ikke null:
 - origin parameter fra HttpRequest er "cartpage" : sendes videre til cartpage.jsp
 - origin = null : sendes til index.jsp
 - Hvis Customer er null: sendes til loginpage.jsp og vises baskedet om fejlen.

Særlige forhold

1. I session gemmes e-mail og customer navn plus en ordre**Ordre er ikke lavet.
2. Sikkerhed ved login: Kodeord bliver gemt i databasen. Kodeord er ikke en attribute af Customer object. Der findes ingen metode som henter kodeord fra databasen.
3. Brugerininput valideres allerede på jsp siden - det er forudbestemt og kræves af input, type i jsp-former
4. Brugertyper: Virksomheder sætter ikke krav til at der skal være flere administratore. Programmet leveres med admin-bruger oplysninger. Administrator findes i users tabellen i database. Programmet validerer email fra logget ind bruger og viser dedikerede funktioner. Programmet udelader Administrator-bruger ved søgning gennem databasen.
5. Exceptions håndteres med try-catch sætninger - der bliver kastet en ny exception ellers findes der nye sætninger til at blive udført, eller med if-else sætning.

Eksempel på "throw new Exception" som exception håndtering:

```

public static Customer login(String email, String password) throws LoginSampleException {
    try {
        Connection con = Connector.connection();

        String SQL = "SELECT firstName, lastName FROM users "
            + "WHERE email='" + email + "' AND password='" + password + "'";
        PreparedStatement ps = null;

        ps = con.prepareStatement(SQL);
        ResultSet rs = ps.executeQuery();

        if (rs.next()) {
            String fName = rs.getString( "columnLabel: \"firstName\"");
            String sName = rs.getString( "columnLabel: \"lastName\"");
            Customer customer = new Customer(email);
            customer.setFirstName(fName);
            customer.setLastName(sName);
            customer.setSaldo(0);
            return customer;
        } else {
            throw new LoginSampleException("Jeg kunne ikke genkende dig. Prøv igen eller registrer dig som ny kunde.");
        }
    } catch (SQLException ex){
        ex.printStackTrace();
        throw new LoginSampleException("Jeg kunne ikke forbindes med databasen. Kontakt butikken");
    }
}

```

Eksempel på "if-else" sætning og kode som exceptions håndtering:

```

if ( password1.equals( password2 ) ) {

    try {

        Customer customer = LogicFacade.createCustomer(email, password1, fName, lName);
        HttpSession session = request.getSession();

        session.setAttribute( s: "email", email);
        session.setAttribute( s: "firstName", customer.getFirstName());
        page = "customerpage";

    } catch (LoginSampleException ex){
        registerMSG="Email findes allerede i vores database. Prøv igen eller log ind";
    }

} else

if (!password1.equals(password2)){
    registerMSG="De to adgangskoder var ikke ens, prøv igen";
    request.setAttribute( s: "givenemail", email);
    request.setAttribute( s: "givenfName", fName);
    request.setAttribute( s: "givenlName", lName);

} else {
    registerMSG="System fejl nr 666. Kontak vnligst butikken: (tlf) 81819292";
}

request.setAttribute( s: "registerMSG", registerMSG);

Order order = OrderLines.getOrder();
int items= order.items();
request.setAttribute( s: "cart", items);
return page;
}

```