

Olsker Cupcakes

Datamatikeruddannelsen

Gruppe 1, hold 20dat2e

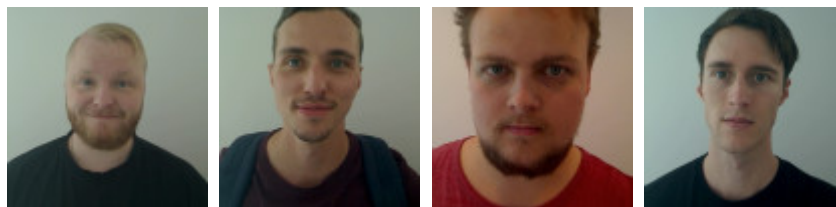
Gruppemedlemmer:

Alex Richardt Gottorp Wagner (GH: A-R-G Wagner)
(cph-aw116@cphbusiness.dk)

Benjamin Choleva (GH: KingBendico)
(cph-bc105@cphbusiness.dk)

Joakim Stensnæs (GH: JoakimKSS)
(cph-js437@cphbusiness.dk)

Lasse Emil Støvring Larsen (GH: Lforlasse)
(cph-ll330@cphbusiness.dk)



Copenhagen Business Academy
24. marts 2020

Abstract

This report documents our first second semester Datamatiker-project at Copenhagen Business Academy. We were tasked with designing and creating a webshop for a cupcake store.

Indhold

1	Indledning	3
1.1	Formål	3
1.2	Baggrund	3
1.3	Teknologivalg	3
2	Krav	4
2.1	Bageriet Olsker Cupcakes	4
2.2	Kravspecification	5
3	Domænemodel og ER-diagram	6
4	Navigationsdiagram	7
5	Sekvensdiagrammer	8
6	Særlige forhold	9
7	Status på implementation	10
8	Test	11
9	Konklusion	12

1 Indledning

1.1 Formål

Vi skal lave en webstore til bageriet Olsker Cupcakes ...

Vi har fået 17 dage (12. marts til d. 29. marts 2020) til at udvikle en webstore til bageriet Olsker Cupcakes. Vi startede med at ...

1.2 Baggrund

Det typiske der skal med for at forklare projektet er:

En kort beskrivelse af den virksomhed, som skal bruge systemet Hvilke krav kunden har til systemet forklaret i brede termer, f.eks. “kunden skal kunne bestille en cupcake hvor man kan vælge både bund og top.” i modsætning til “der skal være en drop down menu med scrollbar i højre side med en liste over hvilke bunde der er og hvad de koster”.

1.3 Teknologivalg

- IntelliJ IDEA 2019.2.3
- JDBC
- MySQL Workbench 8.0 CE
- Adobe XD
- apache-tomcat-8.5.53.

2 Krav

2.1 Bageriet Olsker Cupcakes

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.

Vores opgave er at få udviklet en webstore som skal kunne følgende:

- **US-1:** En kunde skal kunne bestille og betale cupcakes med en valgfri bund og top, sådan at de senere kan køre forbi butikken i Olsker og hente deres ordre.
- **US-2:** En kunde skal kunne oprette en konto/profil for at kunne betale og gemme en ordre.
- **US-3:** En administrator skal kunne indsætte beløb på en kundes konto direkte i MySQL, så en kunde kan betale for sine ordrer.
- **US-4:** En kunde skal kunne se mine valgte ordrelinier i en indkøbskurv, så jeg kan se den samlede pris.
- **US-5:** En kunde eller administrator skal kunne logge på systemet med email og kodeord. Når de er logget på, skal de kunne se deres email på hver side.
- **US-6:** En administrator skal kunne se alle ordrer i systemet, så denne kan se, hvad der er blevet bestilt.
- **US-7:** En administrator skal kunne se alle kunder i systemet og deres ordrer, sådan at denne kan følge op på ordrer og holde styr på sine kunder.
- **US-8:** En kunde skal kunne fjerne en ordre fra sin indkøbskurv, så denne kan justere sin ordre.
- **US-9:** En administrator skal kunne fjerne en ordre, så systemet ikke kommer til at indeholde ugyldige ordrer. F.eks. hvis kunden aldrig har betalt.

2.2 Kravspecification

Vi løser den problemstilling, der beskrives under 2 Case, 2.1: Bageriet Olsker Cupcakes ... For at undgå en mærkelig blanding af dansk og engelsk, har vi valgt udelukkende at skrive vores kode på engelsk.

Dette afsnit vender vi tilbage til senere i semesteret. Der er to dele i dette afsnit:

- Hvad er firmaets håb med dette system (hvad er deres vision for systemet eller hvilken værdi er det jeres system skal tilføre deres virksomhed).
- Scrum userstories (product backlog). Dette vender vi tilbage til!

3 Domænemodel og ER-diagram

Det interessante ved denne domæne og database er at den langt hen af vejen er grundlaget for resten af systemet. Tabeller og relationer siger noget om hvad systemet arbejder med, ikke hvordan. Så det er godt sted at starte.

Som led i beskrivelsen af Domæne- eller ER-diagram skal man have følgende med:

- Diagram over hele modellen. Det er vigtigt at få plads til alle tabeller og alle relationer. Det kan så betyde at man ikke kan få plads til alle attributter på de
- enkelte tabeller. Dem kan man slå op i databasen, så det er ikke så vigtigt
- Hvis nogle af tabellerne ikke er på 3. normal form vil det være almindeligt at nævne det, og forklare hvorfor det er gjort (tidspres eller anden overvejelse).
- Hvis der anvendes 1-1 relationer kan man beskrive hvorfor man ikke blot har en tabel.
- Hvis nogle tabeller implementerer en mange-mange relation vil det være normalt at beskrive det.
- Hvis der er flere veje at nå fra et sted til et andet, vil det nemt gøre det svært at holde databasen konsistent. Hvis I har gjort det alligevel så skal I forklare hvorfor.
- Hvis der er tabeller, hvor man benytter andet end et automatisk genereret ID som nøgle skal man forklare det.

Det er interessant at beskrive hvilke overvejelser der ligger til grund for de konkrete valg der er i ER modellen (fremmednøgler, constraints, triggers, osv.)

4 Navigationsdiagram

Det som brugeren oplever er en række websider, hvor man kan indtaste oplysninger og navigere videre til andre sider. I større systemer kan det være svært at bevare overblikket over hvilke sider der er, og hvordan man kommer fra den ene til den anden. Navigationsdiagrammet er beregnet på at vise dette på en mere overskuelig måde.

Som led i beskrivelsen af navigationsdiagrammet skal følgende med:

- Oversigtsdiagrammet. Hvis det bliver for stort må man dele det op. Men det er vigtigt at der er et overordnet diagram.
- Hvis man har benyttet sig af en “fælles navigations bar” i toppen af alle sider, skal man forklare det.
- Hvis nogle sider kun kan nås af nogle brugere (dem der har konto, dem der er logget ind, dem der arbejder i butikken,...), så skal det fremgå.
- Navne på jsp sider skal fremgå, og hvilke servlets der bringer en fra den ene side til den næste.

Navigationsdiagrammer laves som UML-tilstandsmaskiner state charts. Bemærk specielt at det, der hedder “composite state”, er meget velegnet til at indkapsle at man er logget ind.

5 Sekvensdiagrammer

De fleste programmører kan læse de enkelte metoder i et program, mens det kan være svært at skabe sig et overblik over hvordan programmet virker på overordnet plan. Et sekvens diagram bruges til at vise hvordan et typisk forløb foregår, eller til at vise et særligt svært særtilfælde. Man kan aldrig dokumentere hele programmet med sekvensdiagrammer, man vælger altid nogle interessante eksempler.

Et eksempel på et typisk forløb kunne være at brugeren præsenteres for indkøbssiden. Her skal der vises følgende:

- Selve diagrammet, startende med jsp-siden eller servlet.
- Brug de rigtige klassenavne og metode navne
- Undlad argumenter til metoderne

I forklaringen til diagrammet skal du særligt lægge vægt at beskrive hvilke grene af if-sætninger der er brugt i de enkelte metoder.

6 Særlige forhold

Dette afsnit bruges til at beskrive særlige forhold der benyttes i programmet. Det kan f.eks. være:

- Hvilke informationer gemmes i session
- Hvordan håndterer man exceptions. Det kommer vi tilbage til senere i semesteret.
- Hvordan man har valgt at lave validering af brugerinput.
- Hvordan man har valgt at lave sikkerhed i forbindelse med login.
- Hvilke brugertyper, der er valgt i databasen, og hvordan de er brugt i jdbc
- ... andre elementer

Husk: det er bedre med 2 linjers dokumentation end ingen.

7 Status på implementation

Dette afsnit skal liste hvor langt man er nået med implementationen. Typiske ting man kan have sprunget over er:

- Man har ikke nået at lave alle de jsp sider man har med i navigationsdiagrammet.
- Man har ikke nået at lave alle CRUD metoderne til alle tabellerne
- Man har ikke fået stylet sine sider
- Man har fundet en fejl “i sidste øjeblik”, men har ikke haft tid til at rette det. - F.eks. at man har
- brugt session forkert, sådan at man på en af siderne kan komme ind uden at være logget ind.
- tests der fejler på afleveringstidspunkt
- ... andre mangler

8 Test

Der skal være lavet test. Vi har ikke i dette projekt haft fokus på at teste, så det skal ikke med. Du skal dokumentere tests ved at beskrive i tabel form:

- Hvilke klasser er testet
- Hvilke metoder der er testet
- Dækningsgrad af dine tests for de valgte metoder og klasser

9 Konklusion

Vi har opfyldt minimumskravene ...