Cupcake Projekt 2022

11/4-2022 til 22/4-2022

Klasse A (Gruppe 8)

Navn:	**Cph-email:**	**Github:**
Rehman Abraham	cph-rh255@cphbusiness.dk	https://github.com/notebrr
Oscar Tuff	cph-ot58@cphbusiness.dk	https://github.com/oTuff
Mark Lundgaard	cph-mj924@cphbusiness.dk	https://github.com/Ch0min
Gustav Bøgh	cph-gb84@cphbusiness.dk	https://github.com/GBoegh98

CphBusiness Academy
Cupcake Projekt
11/4-2022 til 22/4-2022

Indholdsfortegnelse

Indledning **2**

Baggrund **3**

Teknologivalg **4**

Krav **5**

User Stories (funktionelle krav) **5**

Modellering **6**

Aktivitetsdiagram **6**

Domæne model **7**

EER diagram **8**

Navigationsdiagram **9**

Særlige forhold **10**

Status på implementation **11**

Proces **12**

Indledning

Denne rapport er dokumentation for vores arbejde med cupcake-projektet. Projektet handler om at kode en simpel dansk webshop til firmaet Olsker Cupcakes ApS, hvor kunderne kan bestille cupcakes for derefter at afhente dem i Olskers fysiske butik. Back-end koden består af Java og koden opdateres på github. Rapporten er henvendt til fagfolk eller studerende inden for IT. Rapporten indebærer diverse diagrammer og forklaring for brug af adskillige redskaber og værktøjer til de forskellige opgaver. Der er blevet arbejdet på kryds og tværs, og planlagt et skema for arbejdsdage på projektet. Vi har kommunikeret med hinanden, for at få det endelige resultat vi kollektivt gerne ville opnå.

Baggrund

Vi er en lille virksomhed fra CphBusiness Academy, som har landet vores første store opgave af en virksomhed baseret på Bornholm, Olsker Cupcakes. Det er et lille nyt grundlagt firma som fokuserer på økologi og kvalitet, og de har ramt helt plet med deres hjemmelavet opskrift, som de nu vil føre ud i livet til andre mennesker. Et par lærere fra skolen har haft snuden forbi bageriet og indsamlet de nødvendige kravspecifikationer, og hermed gennemtænkt og bygget et halvfærdigt mock-up af et muligt design af hjemmesiden.

En mock-up er en løs skitse eller skabelon, som kan give en idé til hvordan en færdig hjemmeside kan komme til at se ud, derfor er det en god kickstarter til designsiden. Der er stadig en masse ting og ideér som ikke er med eller tænkt fuldt igennem, så derfor er det vores tjans, at stille spørgsmål til manglende funktionalitet og design, og komme med forslag m.m.

Vi har tænkt os som leverandør, at løse opgaven på en måde, som deler opgaven op i små bider. Så derfor har vi tænkt i en længere proces om, hvordan vi kan få den første prototype i luften. Derfor skal vi ikke lade os forblænde af den første skitse, og dens funktionaliteter. Vi tager en bid af gangen, og prøver se hvad der kan tilpasses bedst til Olsker Cupcakes krav.

Teknologivalg

Til at kode projektet har vi brugt følgende teknologier, som vi mener er fine værktøjer til de krav som vi er pålagt:

- **IDE: IntelliJ 2021.2.2**
    - Brugt til at kode i. Vi har valgt denne fordi den kan kobles op på git, hvorved vi kan have flere branches. Samtidig har den god intellisense.
- **Java 17**
    - Vi har brugt den nyeste version af Java fordi Java er et populært programmeringssprog og det er nemt at komme i gang med for os da vi i forvejen har erfaring med Java.
- **HTML5 / (JSP)**
    - Vi har brug HTML til at skabe skelettet for vores hjemmeside.
- **CSS 3**
    - CSS har vi brugt til at designe vores HTML skelet, så det føles som et mere fuldendt hjemmeside design.
- **JDBC v8.0.28**
    - JDBC til at koble sql-databasen til java
- **MySQL 8**
    - Mysql til at oprette database. specifikt med Workbench til at forward engineer en database ud fra et EER diagram.
- **Bootstrap v5.1**
    - Bootstrap har vi brugt som et design library, så det giver nogle flotte kanter. Det har hjulpet til design delen.
- **Maven 3.8.5**
    - Brugt til håndtering af dependencies, primært JDBC
- **Git**
    - Git har vi brugt til at arbejde sammen på tværs, så vi hele tiden kunne opdatere hinanden på hvilket nye processer og opdateringer, der er blevet lavet til projektet.

Krav

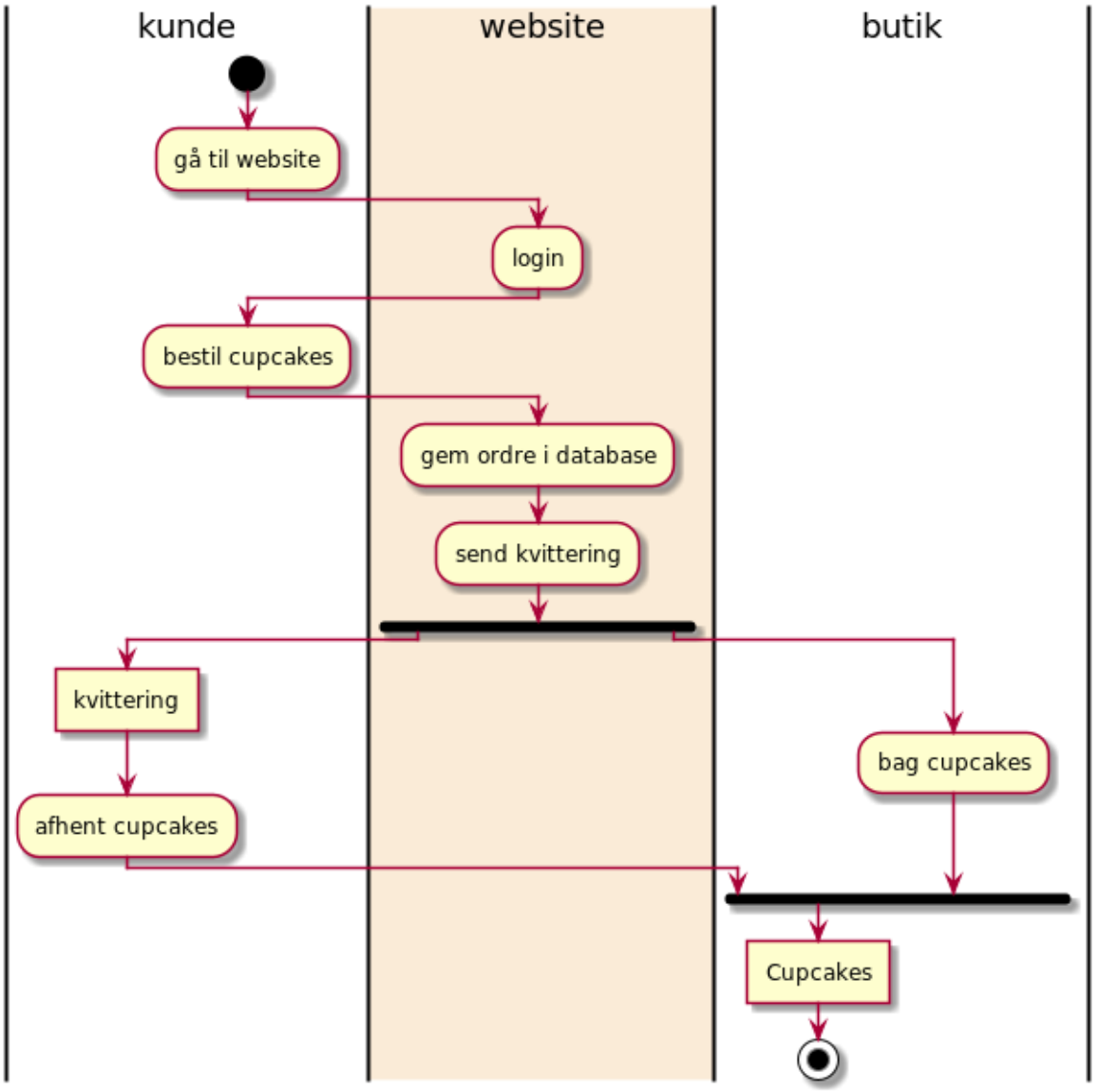
Olsker Cupcakes har en vision og et håb om, at få et overskueligt funktionelt system og dermed også et flot veltænkt design. Hjemmesiden skal være nemt for kunden at navigere rundt i, uden de helt store besværligheder, især i forhold til at bestille en vare eller et produkt. Vi vil gøre det simpelt, men kvalitetsfyldt for Olsker Cupcakes, så de let helt selv kan navigere rundt på siden. Vi vil gøre alt for at hjemmesidens slutprodukt står knivskarpt, og så den i sidste ende opfylder firmaets behov og giver dem en tilfredsstillelse.

User Stories (funktionelle krav)

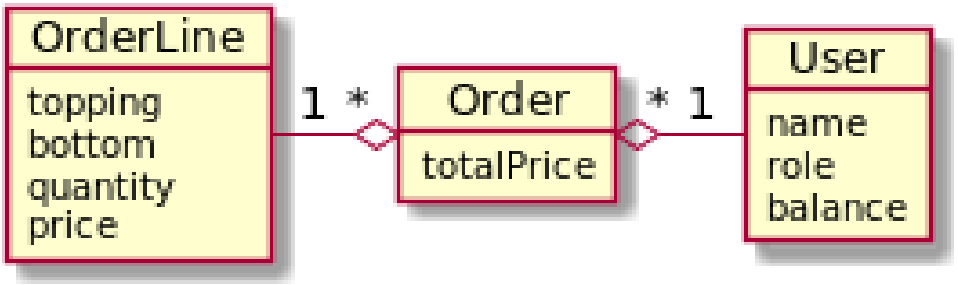
- **US-1:** Som kunde kan jeg bestille og betale cupcakes med en valgfri bund og top, sådan at jeg senere kan køre forbi butikken i Olsker og hente min ordre.
- **US-2:** Som kunde kan jeg oprette en konto/profil for at kunne betale og gemme en ordre.
- **US-3:** Som administrator kan jeg indsætte beløb 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 email og kodeord. Når jeg er logget på, skal jeg kunne se 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. *(Ikke lavet)*
- **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. *(Ikke lavet)*

Modellering

Aktivitetsdiagram



Domæne model

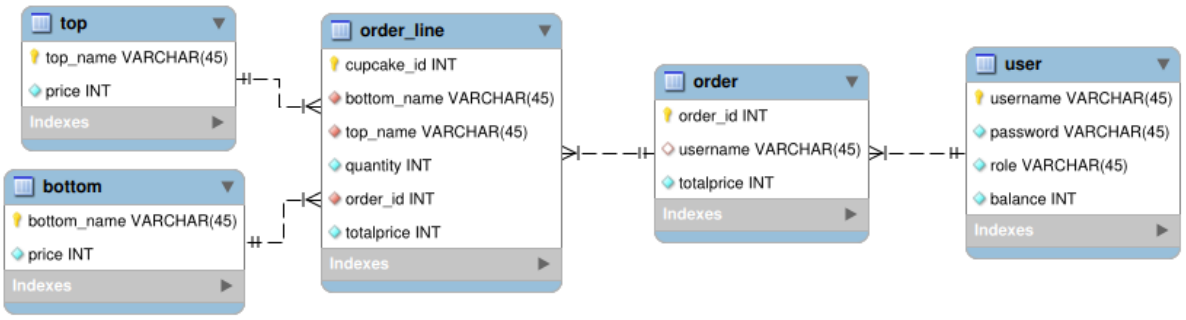


Vi startede med udgangspunkt i cupcake entiteten, som så senere blev til "OrderLine", da det er mere overordnet og at en type cupcake var lig med en "OrderLine".
Vi havde også "User" entiteten fra startkoden.

"Order" bruges til at tilknytte "OrderLine" til "User" med en 1 til mange relation, da en ordre kan have flere ordrelinjer og en bruger kan have flere ordrer.

Attributterne "topping" og "bottom" kunne måske være deres egen entiteter, ligesom i databasen, men under udvikling endte vi med at sætte dem som attributter i "OrderLine" for at gøre det så minimalt og nemt at arbejde med som muligt. De endte som Hashmaps i stedet for deres egen klasse og blev aldrig refaktoreret, da vi fokuserede på andre dele af programmet.

EER diagram



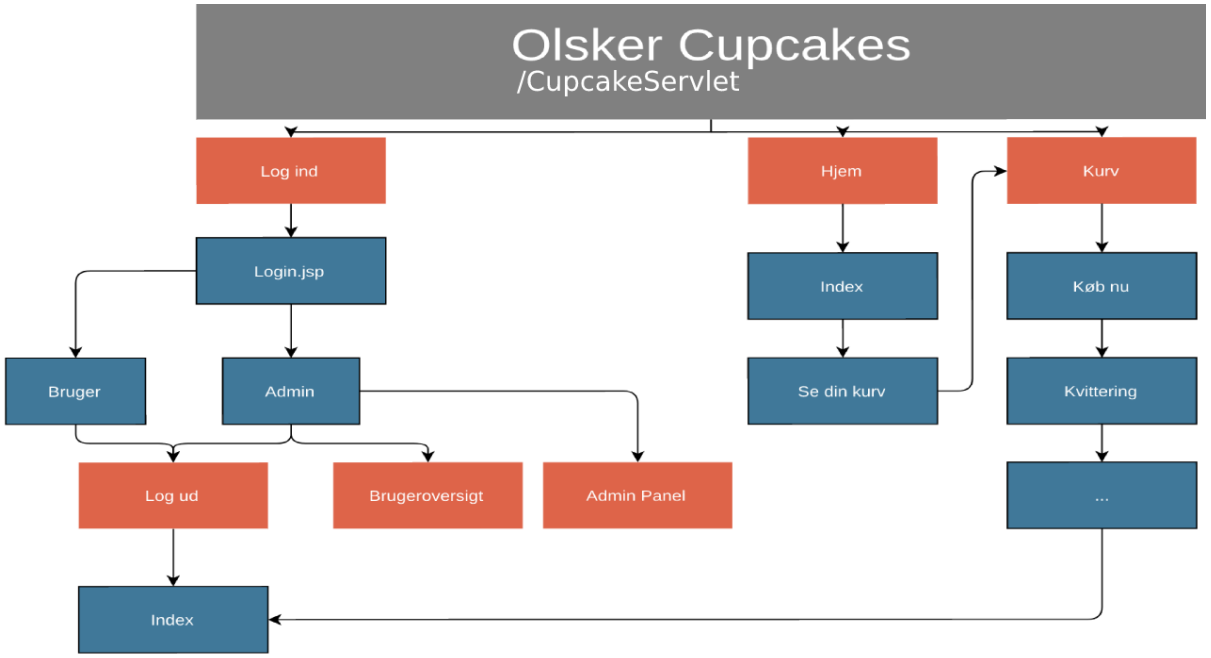
Vi valgte at "top" og "bottom" skulle have henholdsvis "top_name" og "bottom_name" som primary key istedet for id, da de er unikke. Det gjorde det nemmere at arbejde med på java siden af programmet, hvor top og bottom blot hentes som arraylister med hashmaps med name som key, og price som value.

Vi overvejede at tilføje en timestamp attribut til "order", som kunne bruges til at sortere ordre på en admin side med overblik over alle ordrer.
I "order" tabellen er "username" sat til godt at måtte være null i, for at gøre det muligt at oprette ordrer også selvom brugeren ikke er logget ind. Dette er dog ikke implementeret i det endelige system.

"User" tabellen er viderebygget fra startkoden og bruger "username" som primary key, da den er unik. derudover har vi tilføjet "balance", så kunden har en konto til at trække penge fra.

Atributten "role" i "user" tabellen kunne være sin egen tabel for at opnå 3. normalform, men ikke nødvendigt, da der kun er 2 mulige værdier.

Navigationsdiagram



- Orange: Nav-links
- Blå: Nye sider

Vi har valgt at benytte os af en fælles navigationsbar i toppen af alle vores sider, da vi føler at det giver en form for brugervenlighed til brugeren. Det er nemmere at navigere rundt i, når man nemt kan se hvilket links man har i toppen:

- Log ind, Hjem og Kurv

Når man er logget ind som Admin, så har man også adgang til "Brugeroversigt" og "Admin Panel". Dette har man ikke adgang til som almindelig bruger. Når man er logget ind som enten Admin eller bruger, så er nav-link "Log ind" ændret til et nyt link kaldt for "Log ud".

Særlige forhold

I session gemmes et User objekt, og en arraylist som hedder OrderLineList med orderLine objekter.
User objektet bruges til at koble en bruger til en ordre, når dataen gemmes i databasen samt at håndtere brugeradgang til sider ift. om man er kunde eller admin.
OrderLineList bruges til at gemme ordrelinje med info om cupcakes, så man har dem over hele hjemmesiden og først på kurv siden bliver dataene lagt i databasen.
Vi overvejede først at lægge orderLines direkte i databasen, når de blev oprettet, men denne løsning endte med at være mere besværlig og effektiv end vores endelige løsning.

Validering af bruger inputtet på login-siden foregår ved at Java tjekker om der er en bruger med samme email og navn i MySQL via JDBC. På opret bruger-siden smides dataene i databasen uden yderligere validering. På både login- og brugersiden er der validering på front end i HTML, som dog nemt kan forbipasserende. I databasen er koderne gemt uden nogen som helst form for hashing, hvilket gør systemet meget sårbart.

I databasen er der 2 brugertyper: admin og user. I JDBC kan man hive data fra User databasen ned og derefter tjekke den enkelte brugers rolle. Admin-brugeren har, udover at have adgang til de ting user har adgang til, også adgang til en en brugeroversigt side, som viser en oversigt over alle brugere, og en admin side som ikke virker, men hvis formål er at tjekke alle ordrer en kunde har lavet.

Vi har også indført fejlhåndtering på diverse forskellige måder. De fleste løses med simple try, catch statements og til tider throws exceptions. HTTP fejl 404 håndteres ved at vise en fejlside som fortæller brugeren at siden ikke virker.

Status på implementation

På nuværende tidspunkt er størstedelen af hjemmesiden funktionel. Der er enkelte mangler som skal rettes:

- SQL'en i Admin panelet skal tilpasses så den kan hente data på alle ordrer. User story 6 er ikke opfyldt
- User story 9 er ikke opfyldt, admin kan ikke fjerne ordrer som er ugyldige.
- Der er en fejl i opret "Opret bruger" siden, der gør det muligt at indsætte sin egen kode og få den kørt på serveren, via et XSS-angreb
- Der er ikke lavet automatiske tests på funktionerne
- Der er ingen fejlhåndtering af andre fejlkoder end HTTP fejl 404
- Der står øverst på forsiden inden HTML-tagget. Lignende kode kan findes på de andre sider
- Ingen af siderne er mobilvenlige og de er ikke blevet testet på skærme større end 16 tommer
- Ikke testet på alle browsere
- Efter man har klikket på "tilføj til kurv" og dernæst klikker på refresh knappen på browseren, så tilføjer den endnu en cupcake af samme slags.
- På kvitteringen, klikker man ikke på knappen "tøm kurv" men derimod fx "forsiden", så vil kurven ikke blive tømt.
- Brugernes kode gemmes i databasen uden hashing
- Hvis man uden at logge ind trykker på "Køb nu" knappen inde på kurven, vil den vise en fejl
- Ingen CAPTCHA/sikkerhed ved log ind siden eller opret bruger siden - man kan nemt lave en bot til at prøve at cracke folks koder eller til at oprette tusindvis af brugere.

Proces

I vores proces tog vi inspiration fra SCRUM-modellen og delte projektet op i mindre bidder, for derefter at mødes hver dag for at se hinandens kode igennem og opsamle. Helt i starten brainstormede vi de klasser som var de vigtigste og oprettede dem, og delte klasserne op så alle havde noget at lave. Samtidig blev vi også enige om hvordan SQL-databasen skulle se ud. Vi kodede projektet trin for trin, og havde helt overordnet set en ide om resultatet.

I praksis fungerede det godt at dele tingene op i små bider og derefter samle op på det. Det gjorde at ingen af os kom til at spilde for lang tid på unødvendige ting. Det gjorde også at vi kom i mål med størstedelen af hjemmesiden.

Der er også ting som vi i forbindelse med vores proces kunne have gjort det bedre. Vi burde fra start af have brugt længere tid på databasen, fordi vi endte med at ændre på den undervejs, hvilket gjorde at dele af koden i Java og JDBC skulle ændres og at admin-panelet stoppede med at virke, og vi på grund af tidspres ikke fik færdiggjort admin-panelet.

Det vi har lært af processen er helt klart at arbejde bedre sammen i grupper, samtidig med at vi har fået en bedre forståelse for Git. Vi er blevet bedre til at uddelegere opgaver, og hjælpe hinanden med at løse små problemer og bugs. Vi har også hjulpet og lært hinanden de forskellige værktøjer bedre. Det har gjort os bedre som gruppe og til at løse diverse problemstillinger sammen som gruppe.

Til næste gang vil vi gerne blive bedre til at lave en mere struktureret fælles planlægning, og få alle med på samme bølgelængde til hver klasse eksempelvis. Dette ville give en bedre og ligeværdig forståelse af projektet for alle i gruppen.