Olsker Cupcakes
Flow 4 projekt

Projekt information

Projekt titel:

Cupcake

Dato:

26-04-2021

Flow & semester:

Flow 4, Semester 2

Uddannelse:

Datamatiker / Computer Science

Skole:

CPHBusiness Lyngby

Gruppe & klasse:

Group B7, Class B

Gruppe medlemmer, emails & GitHub navne:

Andreas Buch - cph-ab458@cphbusiness.dk - BuchAndreas
Jaan Adam von Magius Butt - cph-jb403@cphbusiness.dk - JaanMagius
Sebastian Fahrenholtz Engelbrecht - cph-se126@cphbusiness.dk - SebastianEngelbrecht
Tobias Jensen Linge - cph-tl233@cphbusiness.dk - Stobes

GitHub projekt:

Link

Indholdsfortegnelse

Indledning

Dette projekt går ud på at lave en simpel webshop. Den skal umiddelbart bruges til salg af cupcakes, men den har også en masse sekundære funktioner, som f.eks. at styre kundekonti og indsætte beløb derpå. Webshoppen skal udarbejdes som introduktionsforløb til vores semesterprojekt, så vi skal benytte os af de samme teknologier og arbejdsmetoder. Det betyder at vi skal bruge en applikationsserver til at hoste siden selv, vi skal designe siden selv ved brug af HTML, CSS & Twitter Bootstrap, og vi skal 'deploye' siden selv ved brug af DigitalOceans droplets. Siden skal naturligvis blot være et proof of concept, men den skal stadig indhente dynamisk data fra vores database, bearbejde denne data med vores backend-logik, og sende det videre til session -og applikationscope. Dette er essentielt en introduktion til et projektforløb for en kunde, som er noget vi kommer til at arbejde meget mere med på studiet.

Baggrund

Cupcake projektet er en opgave som er lavet på vegne af en fiktiv kunde kendt som Olsker cupcakes. De befinder på Bornholm. Vi er blevet bedt om at lave en webshop som kan elevere Olsker cupcakes til nye højder, ved at give cupcake butikken mulighed for at tiltrække nye kunder igennem internettet. Kunderne vil gerne have mulighed for at kunder benytter sig af webshoppen til ordrebestilling, for derefter at hente sin bestilling lokalt i butikken. Kunden vil også have mulighed for at lave kontostyring, sådan at de kan indsætte beløb hos kunder, som de så frit kan bruge i webshoppen. Målet med projektet er at kreere en online del til butikken som kan tiltrække nye kundesegmenter.

Teknologi valg

Vores fællesvalgte IDE som projektet hovedsageligt er udarbejdet i, er IntelliJ IDEA v. 11.0.10. Vi brugte Tomcat Server v. 9.0.44. til at 'deploye' en localhost server, så vi kunne udarbejde websitet. Tomcat projektet bruger JSP, HTML og java dokumenter. Og til at forbinde til vores MySQL Workbench database v.8.0.22. brugte vi JDBC. Vores frontend blev skrevet i HTML og CSS, med hyppigt brug af Twitter Bootstrap v.5.0.0 beta3 til internal HTML styling, så vi ikke behøvede at afhænge af CSS til vores styling. Til slut, har vi brugt Adobe XD v. 38.0.12.13 til at designe mock-ups af sitets udseende, styling, funktionelle knapper og redigering.

Krav

Funktionelle krav

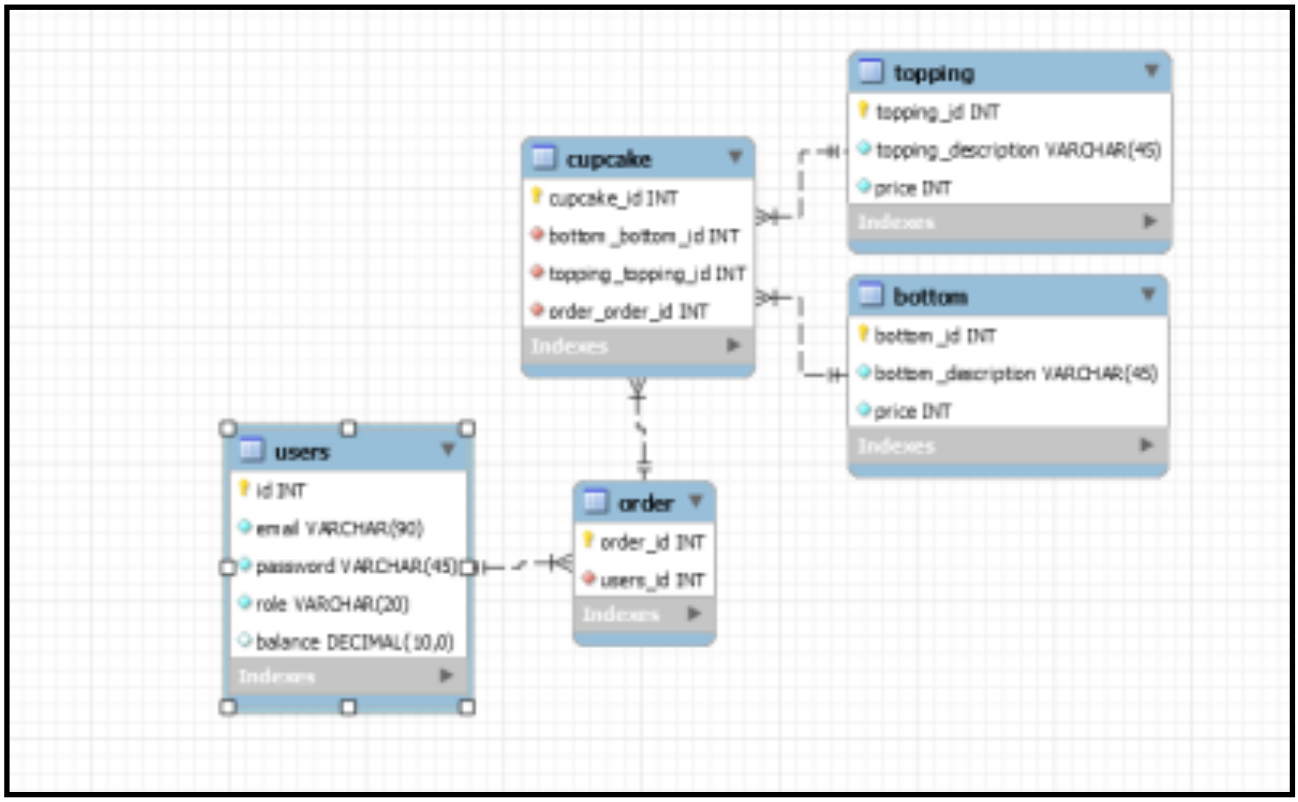
- Som kunde kan jeg bestille og betale cupcakes med en valgfri bund og top, sådan at jeg senere kan køre forbi butikken i Olsker og hente min ordre.
- Som kunde kan jeg oprette en konto/profil for at kunne betale og gemme en ordre.
- Som administrator kan jeg indsætte et beløb på en kundes konto direkte fra MySql, så en kunde kan betale for sine ordrer.
- Som kunde kan jeg se mine valgte varer i en indkøbskurv, så jeg kan se den samlede pris.
- Som kunde eller administrator kan jeg logge på systemet med email og kodeord. Når jeg er logget på, skal min email/brugernavn hele tiden være synliggjort.
- Som administrator kan jeg se alle ordrer i systemet, så jeg kan se hvad der er blevet bestilt.
- 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.
- Som kunde kan jeg fjerne en vare fra min indkøbskurv, så jeg kan justere min ordre.
- Som administrator kan jeg fjerne en ordre, så systemet ikke kommer til at indeholde ugyldige ordrer, f.eks. hvis en ordre aldrig er blevet betalt.

Ikke-funktionelle krav

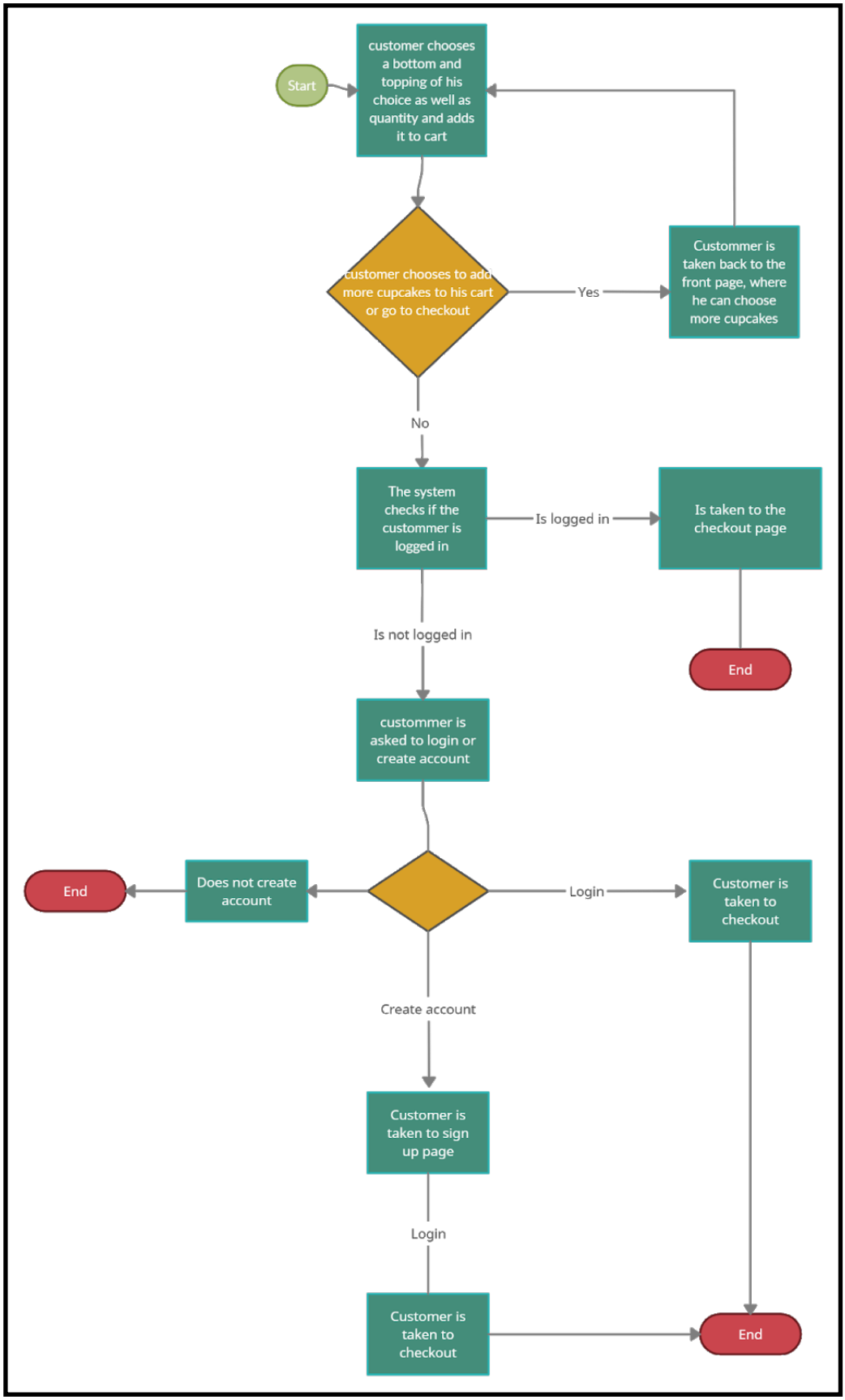
- Der laves en mockup i Adobe XD, som viser de websider den færdige løsning kommer til at bestå af.
- Ordrer, kunder og øvrige data skal gemmes i en database.
- Databasen skal normaliseres på 3. normalform.
- Kildekoden skal deles på GitHub.
- Det færdige produkt skal udvikles i Java, MySql, HTML, CSS, Twitter Bootstrap og køre på en Tomcat webcontainer.
- Det færdige produkt skal i sidste ende køre på en Droplet hos Digital Ocean
- Løsningen skal udvikles med udgangspunkt i vores startkode.

ER Diagram

I vores EER diagram kan vi se hvordan hver user kan lave flere orders (forhold 1 til mange), en order kan have flere cupcakes (forhold 1 til mange), flere cupcakes kan kun have en topping eller en bottom (forhold mange til 1). Efter vejledning med Lars, fandt vi ud af at denne måde at sætte vores EER diagram op på, ville være lettest. Hvis vi havde benyttet os af at lave mange til mange forhold til de forskellige klasser, ville kode delen blive mere kompliceret og tage længere tid at lave. Derudover tog vi et bevidst valg om at indkøbskurven ikke skulle være en del af databasen, da vi ikke ville gemme en halvt færdig ordre i tilfælde af at kunden ikke færdiggjorde købet.

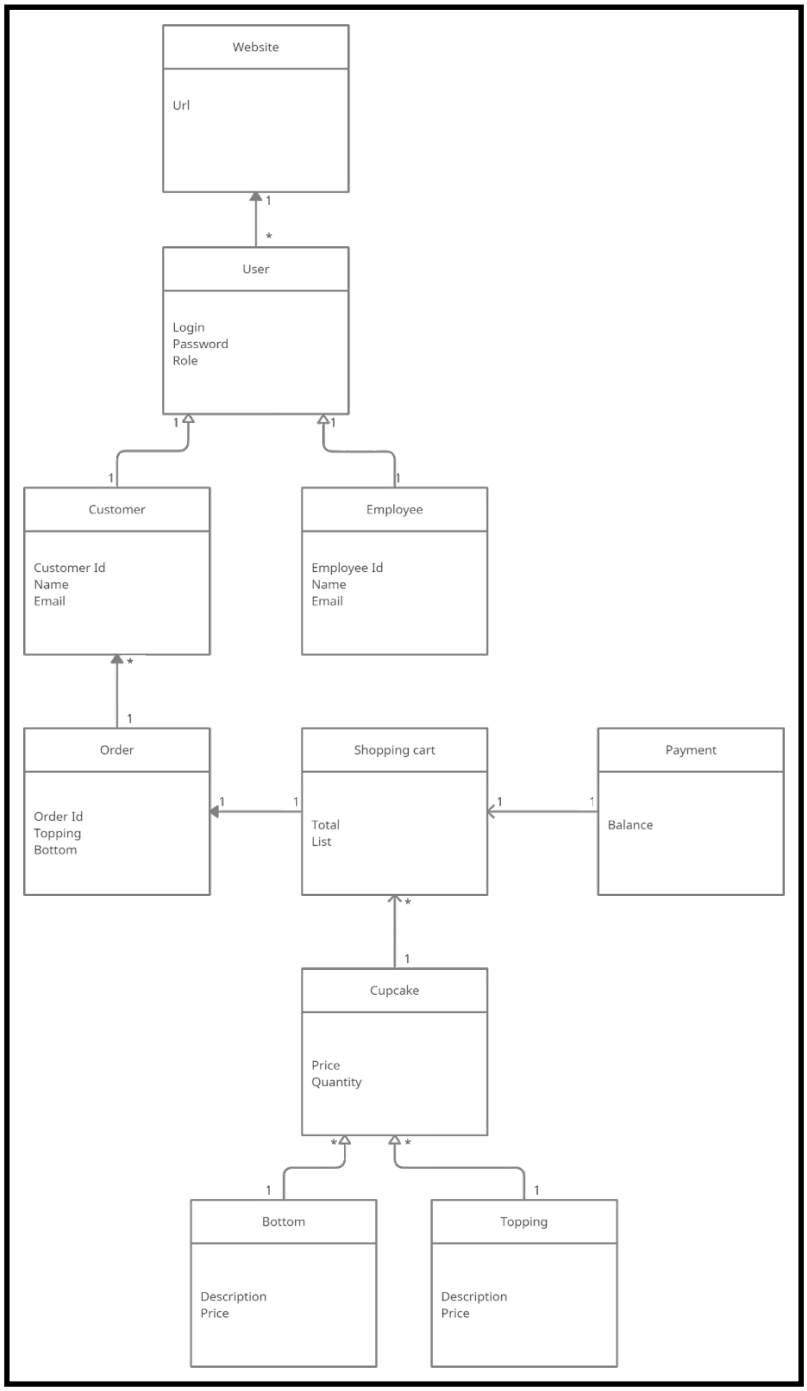


Aktivitetsdiagram



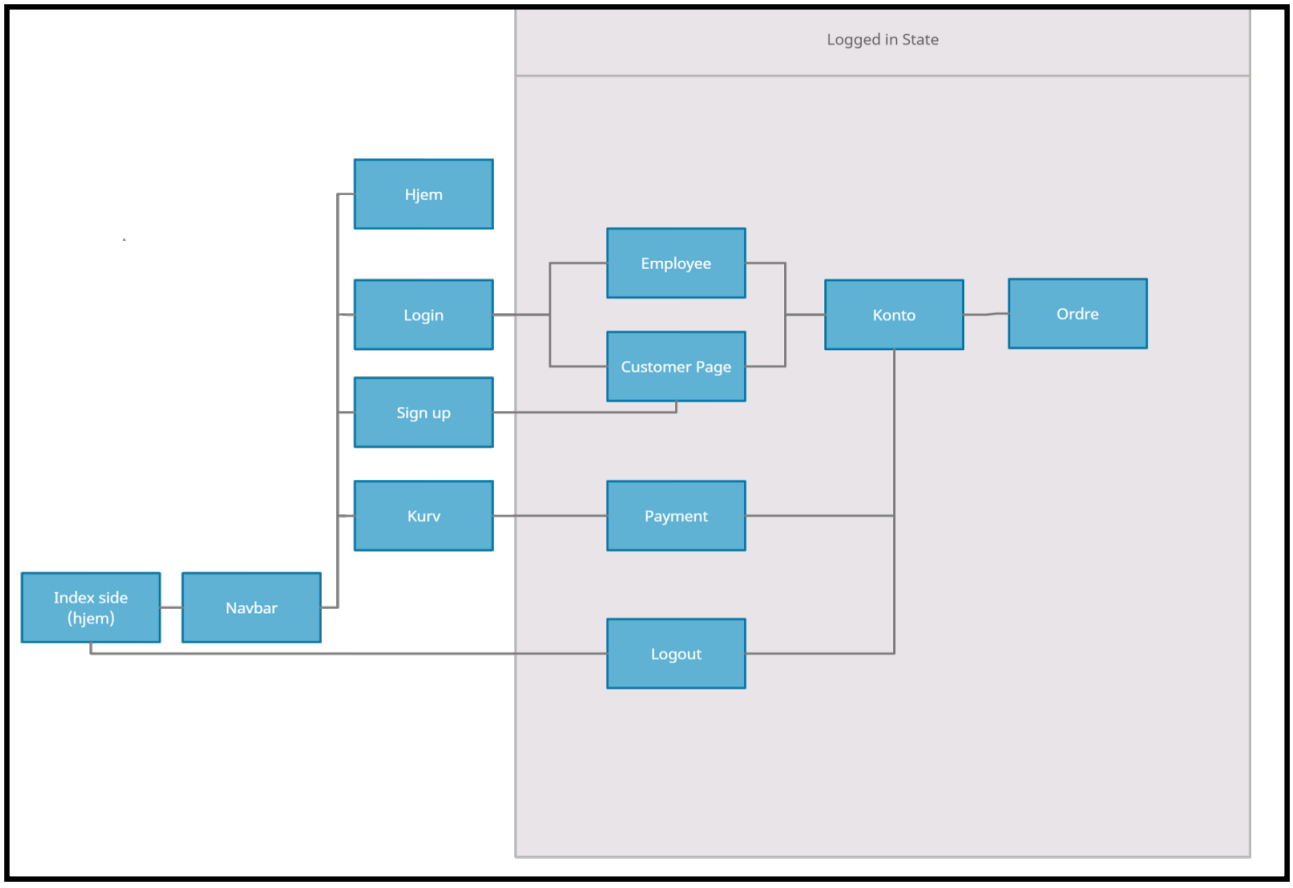
Domæne model

I vores domæne model kan man se forholdet mellem *Users* og *User roles*, man kan derudover også se forholdet mellem et cupcake objekt og *Bottom* og *Topping* klasserne. man kan se at der er et “en - flere” forhold mellem *Shopping cart kl*assen og *Cupcake k*lassen, da en indkøbskurv godt kan have flere forskellige men også af den samme slags cupcake, og det samme gælder *Customer - Order* relationen da en kunde godt kan have flere ordre.



Navigationsdiagram

I navigationsdiagrammet kan man se at vi har valgt at have en navigationslinje på alle sider, hvor at alle elementer på “Navbaren” er tilgængelige for brugere som ikke har “logged in state”.
Vi har også truffet beslutningen om, at for at kunne betale bliver er det et krav at man er logget ind eller opretter en bruger i systemet, og derfor er ordrer også koblet sammen med en konto.



Særlige forhold

Vi har et par særlige forhold i programmet som er værd at gøre opmærksom på. Kigger man på vores login-flow, kan man se det stort set er uændret fra det login flow som vi fik udleveret. Dette inkluderer login-flowets manglende evne til at diskriminere mellem upper -og lowercase. Der var simpelthen vigtigere ting at udarbejde på det tidspunkt i projektet, og derfor fik vi ikke inkluderet det. Til gengæld har vi naturligvis fået benyttet os af protected pages, sådan at vi kan inddæmme præcis hvor vores brugere bevæger sig hen, baseret på deres brugers rolle. Dette gør vi igennem sessionscopet og dets værktøjer. Det betyder vores index side er ubeskyttet og tilgængelig for alle, men de sider hvor der kræves en særlig rolle, eller hvor vi har data som kun må tilgås af en bruger som er logget ind, er beskyttet af protected page. Vores indkøbskurv er også udarbejdet til at benytte sig af session, hvilket tillader os hurtigt og nemt at gemme informationer kortvarigt, uden at skulle oprette poster i databasen. Dette er vigtigt for os, for det betyder at vores kunde ikke skal opbevare data på ordrer som ikke bliver færdig bestilt.

Validering af brugerinput bliver holdt simpelt i vores webshop. Det er gjort ved brug af exception de mere gængse steder, men for områder såsom at vælge bottoms og toppings har vi valgt at bruge drop down menuer frem for input menuer, sådan at vi har fuld kontrol over hvad vores brugere har af input muligheder. Dette hæmmer i stor grad brugerens muligheder for at lave input fejl i et givent input felt.

Status på implementation

De fleste strukturelle elementer af siden er færdigudarbejdede, og det samme gælder for UI og database elementerne. Men vi har stadig nogle mangler som vi ikke fik færdiggjort. Hovedsageligt er der tale om webshoppens købe-flow. Vi har færdiggjort index siden, som tillader en bruger af websiden at vælge en cupcake bund og en cupcake topping. Her kan man så tilføje til indkøbskurven, hvorefter man bliver videresendt til indkøbskurvs-siden; her kan man redigere sin ordre, og gå videre til bestilling, men desværre fik vi ikke færdiggjort logikken til at fuldende sin bestilling, hvilket lige nu resulterer i en fejl. Vi nåede heller ikke at færdiggøre funktionen til at overføre beløb til en kundes konto hvis man var logget ind som en medarbejder. Vi har til gengæld nået at få stylet alle sider vi tager i brug i de forskellige flows, og vi har fået integreret vores html med databasen på de sider hvor det skal bruges.

Proces

Vores gruppes proces har været uformel, men punktlig. Vi aftalte et mødetidspunkt til hver arbejdsdag, som typisk lå omkring klokken 10, hvor alle i gruppen så skulle møde op medmindre andet var aftalt. Når vi så mødte ind blev der kort opsummeret hvor vi nåede til dagen før, og der blev aftalt hvad dagens arbejdsplan gik ud på. Hvilke udfordringer kommer vi til at stå overfor, hvem laver hvad, og hvilke forventninger har vi til de løsninger vi finder på indenfor de specifikke problemområder. Selve arbejdskulturen har været uformel, og det har ikke været unormalt for medlemmerne at måtte pause deres arbejde for at give en hjælpende hånd til en anden fra gruppen for at løse et mere presserende problem med projektet. Vi startede med at udarbejde projektstrukturen og webside sketches som det absolut første, samt en skabelon/mockup til rapporten, så vi nemmere kunne skabe os et overblik over projektets scope. Efter dette begyndte vi hurtigt på kodearbejdet, hvilket udgjorde hoveddelen af dette projekts arbejdsforløb. De tre sidste dage af projektet blev aflagt til rapportskrivning, og til at fikse de absolut sidste småting i projektets kode.

Vi var gode til at holde os til vores tidsplaner, og til at møde op. De mere gængse udfordringer såsom at skabe og style vores JSP sider gik godt og blev hurtigt udformet alt efter vores behov. Det gik til gengæld ikke lige så godt da vi nåede til kodearbejdet med databasen. Her stødte vi på problemer der tog længere tid at løse, særligt hvordan vi skulle befolke vores input menuer med dynamiske data og dernæst sende denne data videre til indkøbskurven gennem session scope. Vi endte med at måtte lave et komplet rework af indkøbskurven på en af de sidste dage for at få det op og køre, hvilket kostede en del arbejdstimer.

Så vi har helt sikkert lært en del programmeringsmæssigt af at løse disse problemer, og vi har ikke haft problemer med gruppemedlemmer som ikke dukker op til tiden, men vi kunne helt sikkert blive bedre til at bede om hjælp når vi løber ind i en mur rent kodemæssigt, i stedet for at prøve at løse det selv og så ende med en masse spildt tid. Det bør vi blive bedre til næste gang, så vi kan udnytte at vi har behjælpelige klassekammerater og lærere. Derudover bør vi som gruppe aflægge mere arbejdstid til projektet næste gang, da man altid vil løbe ind i uforudsete problemer som koster mange timers arbejde.