

Cupcake Projekt

2. semester Datamatiker

Cphbusiness Lyngby
20-04-2022

Simon Lyng Rasmussen - cph-sr297@cphbusiness.dk - GitHub:SimonLyng
Nicklas Vilmann - cph-nv68@cphbusiness.dk - GitHub:Vildmand

Indholdsfortegnelse

Forside	1
Indholdsfortegnelse	2
Indledning	3
Baggrund	3
Teknologivalg	3
Krav	4
Userstories (funktionelle krav)	4
Firmaets håb	4
Aktivitetsdiagram	5
Domæne model og ER diagram	5
Domænemodel:	5
EER diagram	7
Navigationsdiagram	8
Særlige forhold	9
Status på implementation	9
Proces	10

Indledning

Denne rapport er skrevet, så en studerende på 2. semester kan læse den. Der vil derfor være begreber, som den typiske person ikke kender.

Dette projekt omhandler at lave en dynamisk hjemmeside for Olsker Cupcakes. Hjemmesiden skal have bestillingssystem, Databaser over brugere og cupcakes.

Man skal kunne oprette nye brugere, og kunne logge ind på nuværende kontoer.

Derudover skal hjemmesiden have en Admin-bruger som har rettigheder, til at se alle ordrer og ændre i dem.

Baggrund

Olsker Cupcakes er et bageri på Bornholm, som har ramt den helt hellige opskrift. Bageriet er et økologisk iværksættereventyr, som får mange gæster fra hele landet. Derfor søger Olsker Cupcakes nu en ny hjemmeside, som kan holde til alle deres nye kunder.

- Kunden skal kunne bestille en cupcake hvor man kan vælge både bund og top
- Kunden skal kunne oprette en ny konto for at kunne betale/gemme end ordre
- Kunden skal kunne se sine valgte ordrelinjer i en indkøbskurv, samt se en samlet pris
- Kunden skal kunne fjerne ordre fra indkøbskurven, eller justere i ordren
- Kunden skal have en administrator-bruger, som har rettigheder til alt
- Administrator-brugeren skal kunne se alle ordrer som er blevet bestilt
- Administrator-brugeren skal kunne fjerne og justere ordrer

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”*.

Teknologivalg

IntelliJ 2021.3

MySQL Workbench 8.0

Tomcat 9.0.60

Krav

Userstories (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 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 mockup'en).

US-6: Som administrator kan jeg se alle ordrer i systemet, så jeg kan se hvad der er blevet bestilt.

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.

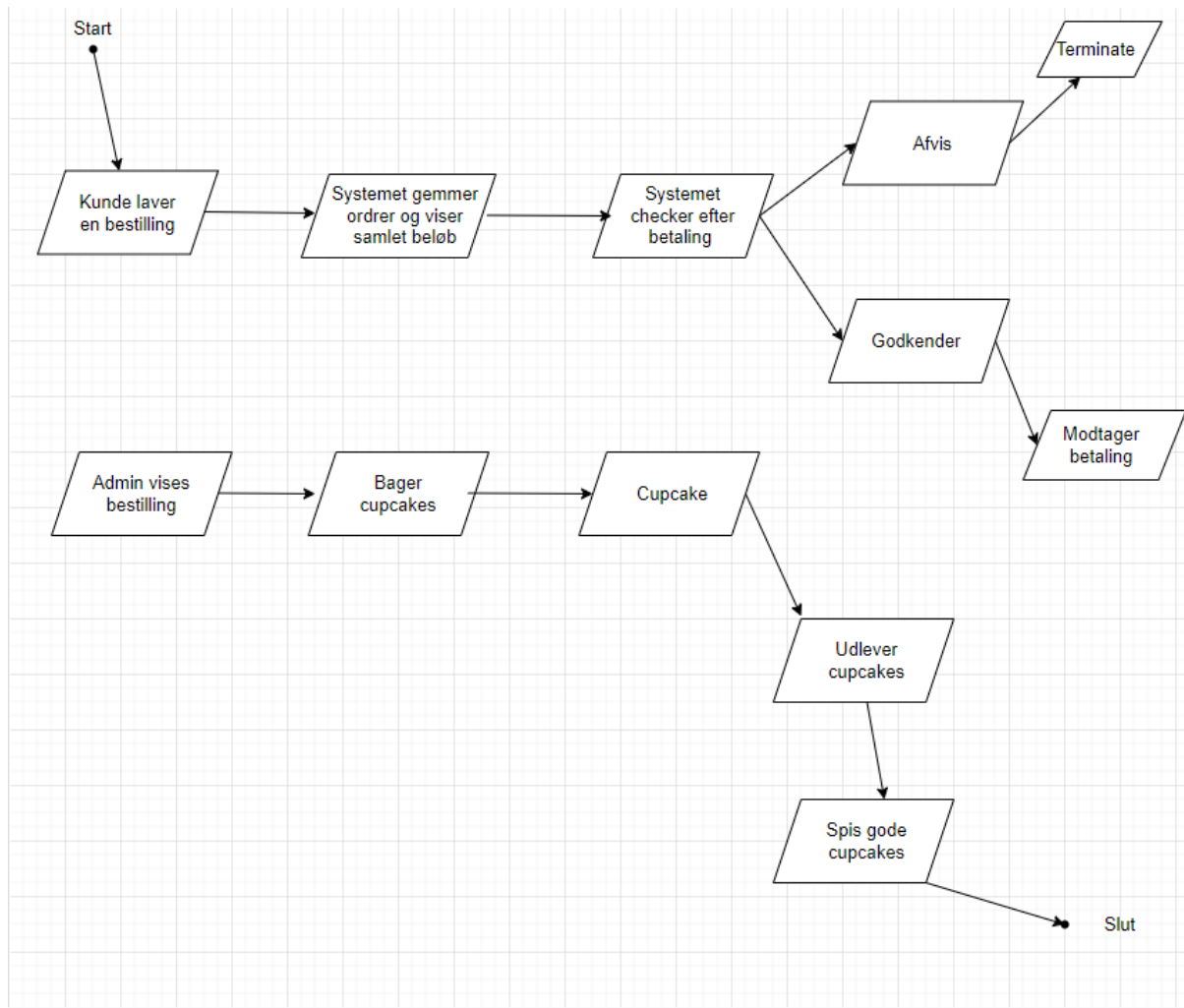
US-8: Som kunde kan jeg fjerne en ordre fra min indkøbskurv, så jeg kan justere min ordre.

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.

Firmaets håb

En kørende website, som løser behovet som blev stillet til mødet. Indtil videre skal websitet blot være på egen laptop, uden at være på en virtuel maskine. Firmaet vil se, at som minimum bliver de seks første Userstories udført.

Aktivitetsdiagram



I dette aktivitetsdiagram kan man se, hvordan processen foregår. Det start med, at kunden laver en bestilling, hvor den flyver igennem systemet, og enten betales eller afvises. Hvis ordren afvises, bliver hele processen afsluttet. Hvis der bliver betalt, kan Admin-brugeren se, at der skal laves antal cupcakes, som skal udleveres til kunden. Så kan kunden nyde sine cupcakes, og processen bliver afsluttet.

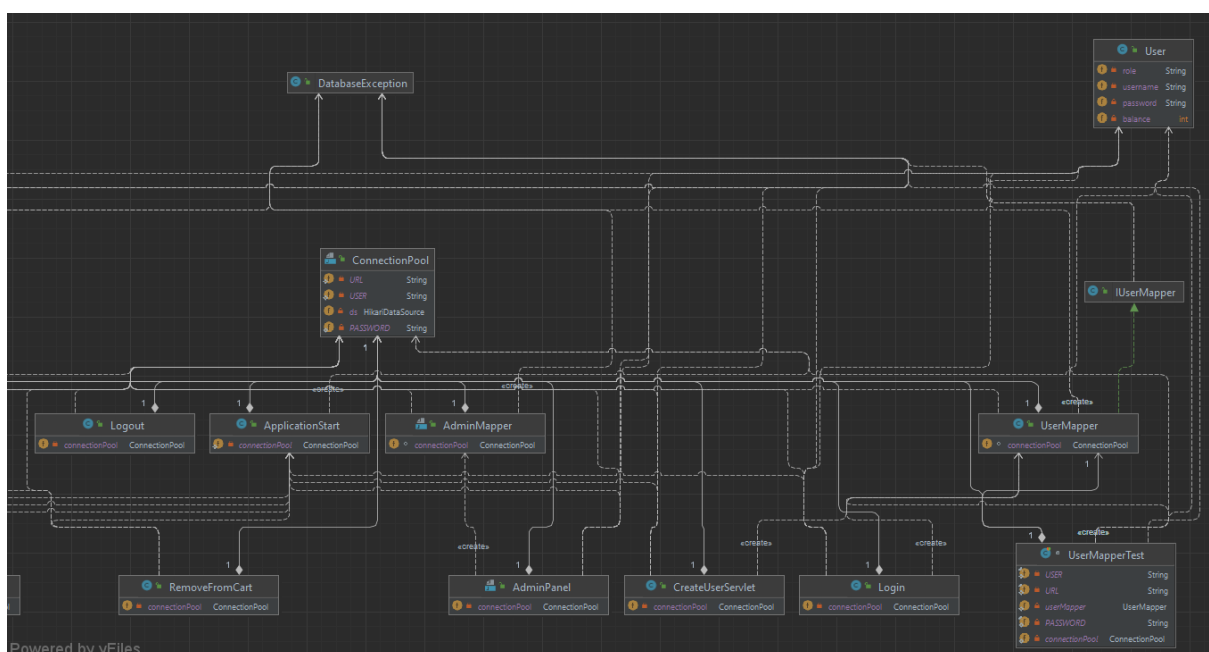
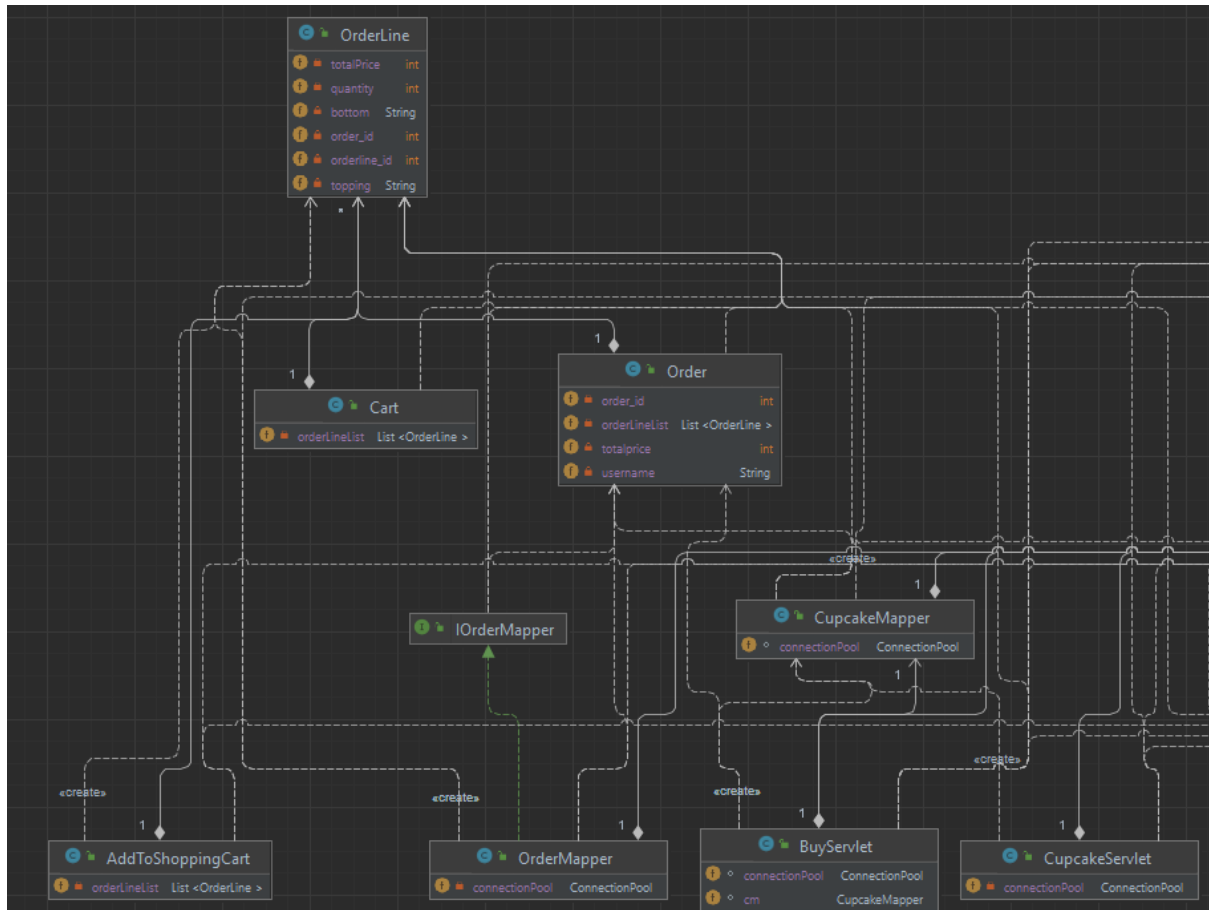
Domæne model og ER diagram

Domænemodel:

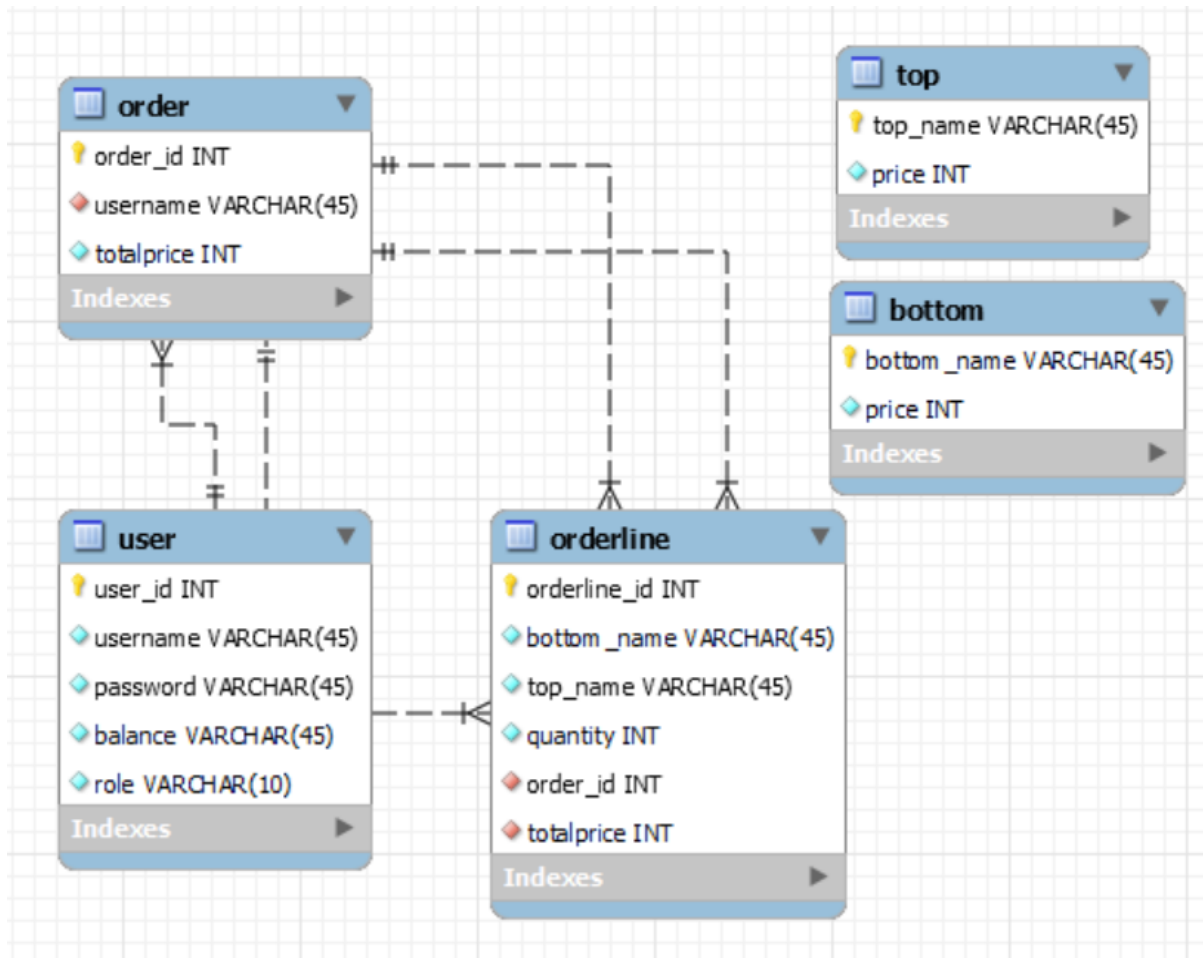
kan ses under src->resources->cupcake.uml

Vores største overvejelser i relationerne mellem klasserne er især hvordan vi skulle opsætte "order" og "orderline" samtidig med at få det til at spille med "cart". Vi snakkede meget om

hvilken af klasserne der skulle "samle" alle klasserne. Til at starte med tænkte vi at "order" var bedst da vi i vores database havde "order" som styrende tabel. Tilføjelsen af "cart" ændrede dog planen lidt da man både skulle kunne tilføje og fjerne fra kurven og derfor også fra ordren. Relationen gav derfor bedre mening hvis vi valgte at det hele sluttede med "orderline" da det gav mere frihed til "cart".



EER diagram



Som udgangspunkt var ideen at bygge databasen på 3. normal form. Hvilket vi også i store mængder af processen har overholdt. Kan fx ses i "order" og "orderline" hvor vi har valgt at splitte disse 2 ud i hver deres tabel for at holde de attributter der ikke har en direkte tilknytning til primærnøglen i tabellen, ud i deres egne nye tabeller.

Alle tabeller i databasen som har et ID er automatisk generet så vi er sikker på at alle de ID's der bliver givet, er et unikt nummer. Dette sikrer at vi nemt kan holde styr hvilken kunde der har bestilt, hvilket ordre kunden har osv.

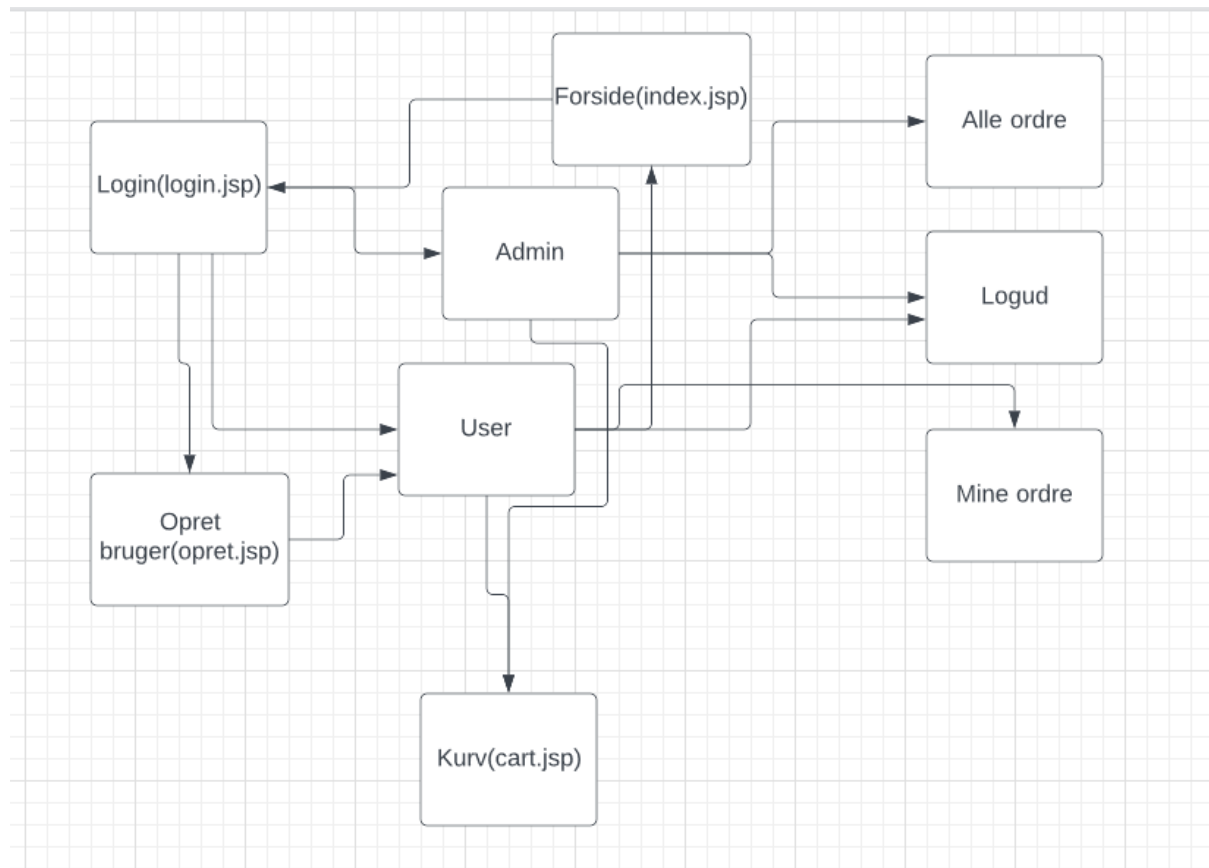
Vores overvejelser bag resten af vores fremmednøgler er først og fremmest at få en kontinueret forbindelse mellem tabellerne. Det er bygget op så en administrator på hjemmesiden kan få alt det "vigtige" information fodret til sig med det samme. Vi samler

derfor det meste i "order" administrator kan se hvem og hvad der er blevet bestilt og derfor hurtigt danne overblik.

Databasen slutprodukt er meget mindre end hvad vi startede ud med. Vi havde tabeller som "Products" og "Admin" som enten var for overflødig eller blev iscenesat på en anden måde. Som det kan ses i EER diagrammet har products udviklet sig til 2 tabeller henholdsvis "top" og "bottom". Dette blev gjort af praktiske årsager. Det gjorde det ganske enkelt nemmere for os at få de valgmuligheder som man skal have af toppings og kagebund som kunde, ind på hjemmesiden. Det kan ses i EER diagrammet og vores mangel af fremmednøgler at vi har været under tidspres da denne ændring blev lavet. I "orderline" har vi en tydelig forbindelse til "top" og "bottom" men for ikke at sætte forbindelse mellem de tabeller op. Det er noget vi gerne ville have gjort anderledes men mangel på tid og jagten på en løsning på vores problem satte os med andre prioriteter.

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:



Særlige forhold

- Email gemmes i session, når man logger ind
- Log ud gemmes i session, når man logger ind
- Konto gemmes i session, når man logger ind
- Når man logger ind, tjekkes der om kontoen findes
- Hvis konto ikke findes, sendes man til error page for at oprette en konto
- Kurven gemmes når man er logget ind
- Admin kan se alle ordrer som er blevet gemt(købt)
- Admin kan se alle kunder som har en konto

Status på implementation

- Manglende jsp side til ordrebekræftelse når man vil købe sine vare.
- Fejl når man prøver købe varer. Kurven bliver ikke reset og varerne bliver i kurven.
- Kundens ordre bliver ikke gemt i databasen

Proces

Gennem projektforløbet har vores planlægning været meget ustabil. Dette skal ikke forstås som om vi ikke har planlagt projektet eller sat delmål for processen. Vi har stødt på mange forhindringer på vejen hvilket har resulteret i at vi måtte omlægge planer og den tid der var sat af til den del af projektet. Da vi startede projektet blev vi enige om at vi ville sætte delmål på hvad der skulle laves på en dags arbejde. Vi satte os derfor altid ned og snakkede om hvad der var agenda for dagen og hvad vi manglede eller havde nået. Da vi kun var 2 var det en meget nem samtale da vi begge havde et godt overblik over hvad den anden havde lavet dagen før. Generelt var kommunikationen god. Grundet den gode kommunikation fra starten af følte vi at opdeling af arbejde ville være optimalt. Vi holder konstant hinanden opdateret på vores arbejde men sad altså ikke og arbejde på den samme ting medmindre der var brug for assistance.

Vores arbejdsform virkede helt fint i praksis og vores aftale om at holde møder hver morgen og ligge mål for dagen blev også overholdt gennem hele projektforløbet. Vores måde at opdele arbejdsopgaver blev dog ikke holdt hele vejen igennem da vi hurtigt støtte på andre problemer som gjorde det nødvendigt at sidde sammen om løse forhindringen.

Der var både op og nedture i processen. Som sagt gik det rigtig godt med at kommunikere konstant og ikke lade den anden falde bagud på information. Det gjorde det processen mere flydende og selvom der opstod problemer så var det ikke kun 1 person der sad med det da der hurtigt blev kommunikeret og sat fokus på at få løst problemet. Det gjorde at vi hurtigere kom videre i processen og ikke sad fast alt for længe. Processen var ikke uden problemer da vi fik et problem der var utrolig svært at løse. Grundet problemer med Maven filer sad i fast i et godt stykke tid, og vores kompetencer var ikke stærke nok til at løse problemet selv. Det kunne godt mærkes på gruppen at vi var irriteret over vores manglede værktøjer til at kunne forstå og løse vores problem vi havde ved hånden. Vi kunne i den situation godt have været bedre til at ligge vores tro på individuelle evner væk og søgt hurtigere assistance fra en rådgiver da det var en mulighed at få hjælp. Det kunne mærkes at vi havde siddet med problemet længe og derfor var blevet halv irriteret på hinanden selvom det ikke var et problem der kunne fikses af os på stående fod. En prioritering af gruppen morale ville have været en bedre beslutning.

i løbet af projektet og den proces vi har været igennem har ikke forladt os tomhændet. Vi har lært at ligge egoet til siden og spørge efter hjælp ikke skader, selvom det gør ondt i momentet. Derudover har vi fået lært at en projekt ikke bare laves som sådan. Det er en proces, en rejse man skal igennem. Der er gode timer og mindre gode timer men det er alt sammen en del af forløbet.

Noget vi kunne have gjort anderledes i projektet er vores gruppestørrelse. Den korte tid vi havde til at få projektet gjort færdigt og flot er ikke egnet til kun 2 personer. Vi fortryder det ikke, da vi har lært utrolig. Det var dog hårdt og et ekstra medlem ville have gjort forløbet nemmere. Især når det kom til både at få en backend og frontend op og køre som 2 var svært. Backend var klart det der skulle lægges mest tid i for at kunne nå at overholde kundens ønsker. Så prioriteten blev også lagt der det meste af tiden men en ekstra par hænder have sikret at ihvertfald 1 person kunne være på frontend konstant. Ikke desto mindre står vi tilbage med et helt fint projekt på kort tid og arbejdskraft det er ikke perfekt og mange ændringer kan foretages i fremtiden. Processen og projektforløbet har efterladt os en hel del klogere end da vi startede