



Cupcake projekt

Udarbejdet opgave: 19 april 2021 til 26 april 2021

Udarbejdet rapport: 19 april 2021 til 28 april 2021

Informationer

Navn: Alex Simone Kenfelt

Mail: cph-ak428@cphbusiness.dk

GitHub: <https://github.com/AlexKenfelt>

Klasse: A klasse

Navn: Maja Nøhr-Nørgaard

Mail: cph-mn605@cphbusiness.dk

GitHub: <https://github.com/cph-mn605>

Klasse: A klasse



Indhold

Informationer	1
Indledning.....	3
Baggrund (for opgaven):	3
Teknologivalg.....	4
Mockup:.....	5
Diagrammer	6
EER:	6
Navigationsdiagram:.....	7
Domænemodel:.....	8
Aktivitetsdiagrammet:	9
Særlige forhold	12
Status for implementation.....	12
Proces	13

Indledning

Vi har fået til opgave, at lave en webshop for den Bornholmske cupcake forretning Olsker, hvor en kunde selv kan sammensætte, bestille og betale for sin cupcake.

Websitets design er taget udgangspunkt i et mockup skabelon, som er blevet lavet ved hjælp af Adobe XD. Skitsen skal give en ide, om det færdige produkt og det vi vil læne os op ad.

Websitet bliver primært kodet ved hjælp af HTML, CSS, Bootstrap, Java og SQL, som bliver kørt igennem Tomcat webcontaineren, til at eksekvere programmet. Websitet er derfor dynamisk, da der kan gemmes og hentes data fra en database, som bliver normaliseret på 3. normalform.

Websitet skal kunne tilgås på to måder, der skal være en kunde-og en administratorindgang. Når brugeren er logget på, kan man se brugerens email i headeren under hele sessionen.

Når kunderne logger ind på Olskers webshop, skal de kunne oprette en profil, sammensætte deres ordre af cupcakes. Ordren består af, at kunden kan vælge mellem forskellige bunde, toppe og antal af cupcakes. Kunden skal derefter kunne lægge dem i en indkøbskurv. Indkøbskurven bliver vist som en ordrelinje, samt et samlet beløb for ordren, som kunden derefter skal kunne betale. Kunden skal derudover også have mulighed for, at kunne slette en ordre fra sin ordrelinje.

Baggrund (for opgaven):

Vi har lavet en opgave for Olsker Cupcakes, som har oplevet det største iværksættereventyr på Bornholm, da de har fundet den helt rigtige opskrift på cupcakes. Nogle hipstere fra København har igangsat en mock-up, som skitser løst hvordan webshoppens færdige resultat skal se ud. På det grundlag har vi talt videre med dem, om hvordan den færdige mockup skal se ud, så vi kan arbejde ud fra den, samt hvilke funktionaliteter de forventer webshoppens kommer til at have. Målet er at møde deres behov.



Ud fra mødet med kunden, har vi udarbejdet følgende user-stories, for at kunne følge kundes krav bedst muligt:

NR	User stories	Status
US-1	Som kunde kan jeg bestille og betale cupcakes med en valgfri bund og top, sådan at jeg senere	Mangler betaling
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.	Funktion er lavet
US-4	Som kunde kan jeg se mine valgte ordrelinier 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 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.	

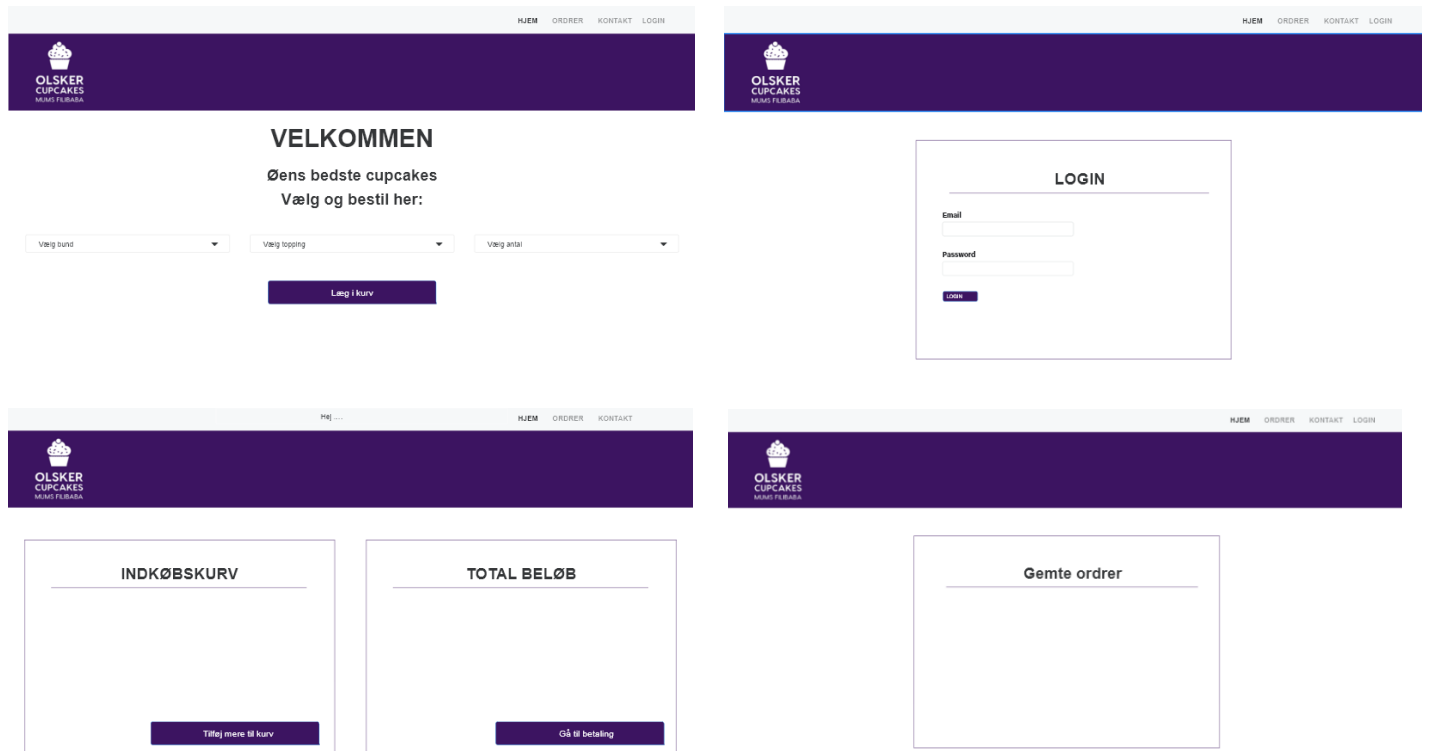
Teknologivalg

Vi har brugt følgende teknologier til udarbejdelsen af projektet:

- Adobe XD
- Draw.io
- Discord
- Git Bash
- Git Hub
- IntelliJ
- Java 8
- Google Chrome
- Microsoft Edge
- MySQL Connector/Workbench
- Tomcat
- Word
- Windows
- Zoom

Mockup:

Baggrunden for websites design, er lavet ud fra følgende mockup:

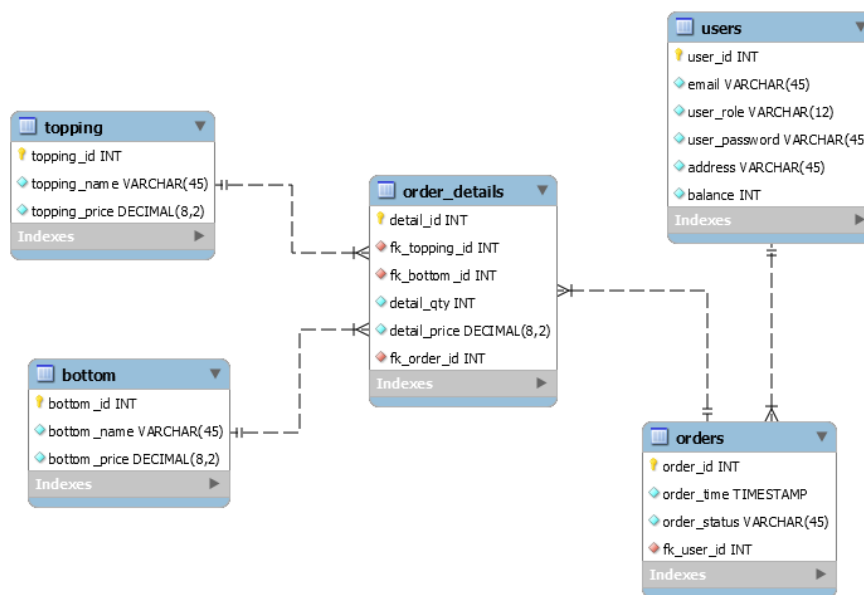


Mockuppet blevet lavet i startfasen for projektet, så alle involveret i projektet ville have en ide om, hvordan det færdige resultat ca. vil se ud. Mockupen er ikke fuldt til punkt om prikke, men layoutet for websitet er blevet holdt nogenlunde.

Diagrammer

EER:

Databasen er designet med tabeller og relationer, der næsten overholder 3. normalform. Det vil sige, at databasens kolonner er ikke-transitivt afhængige af en primærnøgle. Afvigelserne fra 3. normalformen sker i user_roles, da der godt kan fx være flere kunder. Man kan også være udenfor der kan være den samme data mht. users adresse, grundet to kunder godt kan leve på samme vej eller endda indenfor samme husstand. Afvigelsen giver mening at lave, da det vil gøre det mere uoverskueligt, at tilgå databasen, med tables kun for overnævnte.



ER-diagram består af fem tables: users, orders, orderdetail, topping og bottom.

Vi har en primærnøgle (PRIMARY) i hver table, som definerer den tabel. Primære nøglen er auto increment i alle tabellerne. Der bliver derfor skabt en unique nøgle i den enkelte tabel.

Fremmednøgler (FK) bliver også brugt i et eller flere felter i tables. Det identificerer en række i et andet table, det kalder man for børnebordet, som bruges til at henvise til forældre bordet.

User indeholder:

- user_id (PRIMARY), email, role, password, address og balance.

Orders indeholder:



- orders_id (PRIMARY), order_time, order_status, (FK) user_id.

Orderdetail indeholder:

- detail_id (PRIMARY), (FK) topping_id, (FK) bottom_id, detail_qty, detail_price, (FK) order_id.

Topping indeholder:

- topping_id (PRIMARY), topping_name, topping_price.

Bottom indeholder:

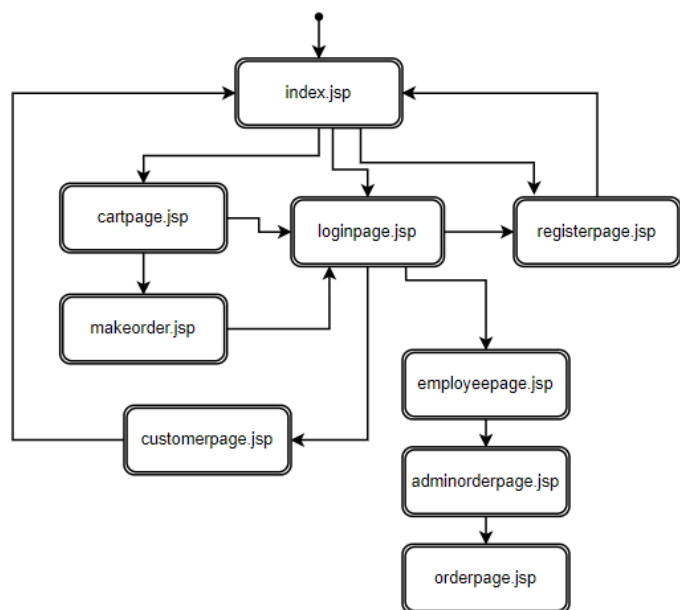
- bottom_id (PRIMARY), bottom_name, bottom_price.

Relationer mellem tables:

- Der er en til mange relation mellem users og orders.
- Der er en til mange relation mellem orders og orderdetails.
- En til mange relation mellem topping og orderdetails.
- En til mange relation mellem bottom og orderdetails.

Navigationsdiagram:

Navigationsdiagrammet er med til at vise, hvordan man navigerer rundt i sit system. Den sorte prik indikerer, at det er her vores program starter. Når vi rammer index.jsp som er vores “forside” får man mulighed for at bestille cupcakes. Der er lavet knapper, som gør man kan vælge at gå til cartpage.jsp, som er vores “indkøbskurv”, login.jsp og registerpage.jsp som gør det muligt at





oprette sig som bruger. På index.jsp har man mulighed, for at bestille sine cupcakes og overføre dataerne til sin indkøbskurv.

På cartpage.jsp er der blevet lavet to veje, en til hvis man ikke har logget ind og en til hvis er logget ind. Hvis man ikke er logget ind, så bliver man sendt over til login.jsp med de gemte dataer. Når man er logget ind og trykker “Bestil” bliver man smidt over til makeorder.jsp som er vores “kvitteringsside”. Dataerne bliver smidt i vores database som kan overføre dataerne til vores administrator bruger.

Som administrator, starter man ude på employeepage.jsp. Her er lavet en knap, som kan sende administratoren videre til adminorderpage.jsp. Der kan administratoren, udskrive ordrer og fjerne ordrer.

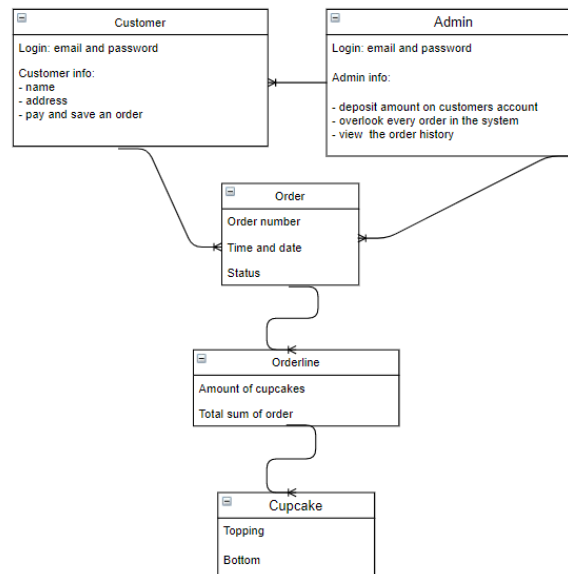
Når kunden logger ind, bliver vedkommende videreført over til customerpage.jsp. Her er der oprettet en knap, som sender vedkommende hen til index.jsp.

Domænemodel:

Kunden ønsker en webshop, hvor brugeren kan sammensætte, bestille, oprette og betale for sine cupcakes. Brugeren skal derfor også kunne oprette sig via sit navn, e-mail og et password. Udover en brugerflade bliver der også en administrator indgang, så Olsker har mulighed, for at administrere systemet og se orde historikken for salg af cupcakes.



Domænemodel



Domænemodellen består af tabellerne:

Customer, Admin, Orde, Orderline og Cupcake.

Relationen mellem tabellerne er:

- Kunden til ordre er en til mange relation, da en kunde kan afgive flere ordre.
- Ordre til ordrelinjer har en til mange relation, da en ordre kan indeholde mange ordrelinjer.
- Admin og customer har en til mange relation. Da vi har en admin, som kan se mange kunder.
- Admin og orders/orderline har en til mange relation. Da admin kan se alle ordre.

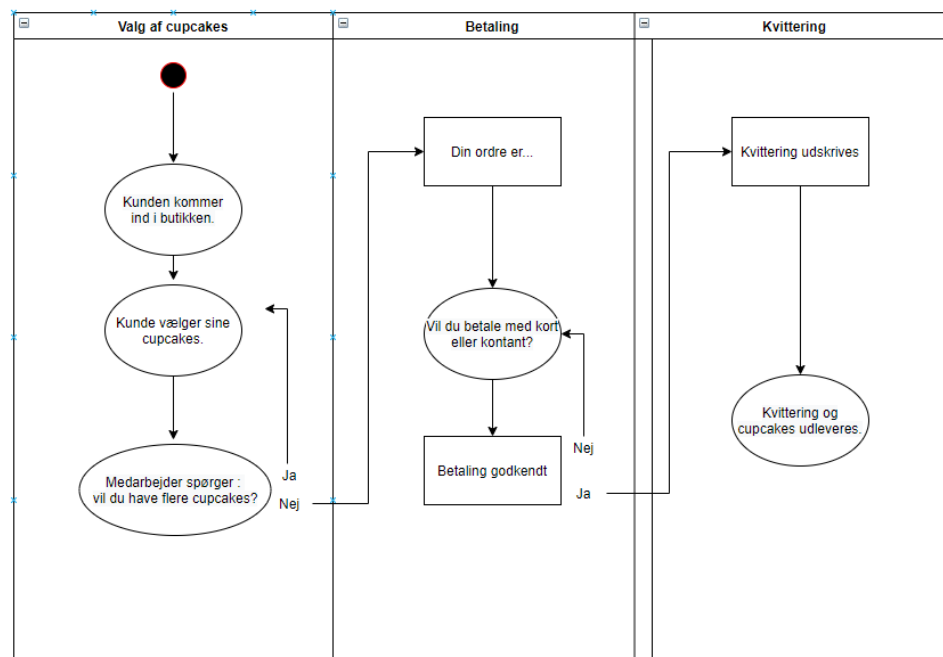
Aktivitetsdiagrammet:

Vi har lavet to forskellige aktivitetsdiagrammer. Et der viser hvordan butikken fungerer fysisk, samt et der viser hvordan det kommer til at fungere via webshoppen.



As-is:

Kunden starter, med at komme ind i butikken. Kunden bliver spurgt af ekspedienten, hvilke cupcakes kunden kunne tænke sig? Hvis kunden er færdig med sin bestilling, gentager ekspedienten ordren og spørger om kunden vil betale med kort eller kontant. Hvis betalingen bliver godkendt, udskrives en kvittering fra kassesystemet og kunden får udleveret sin kvittering og sin cupcakes ordre.

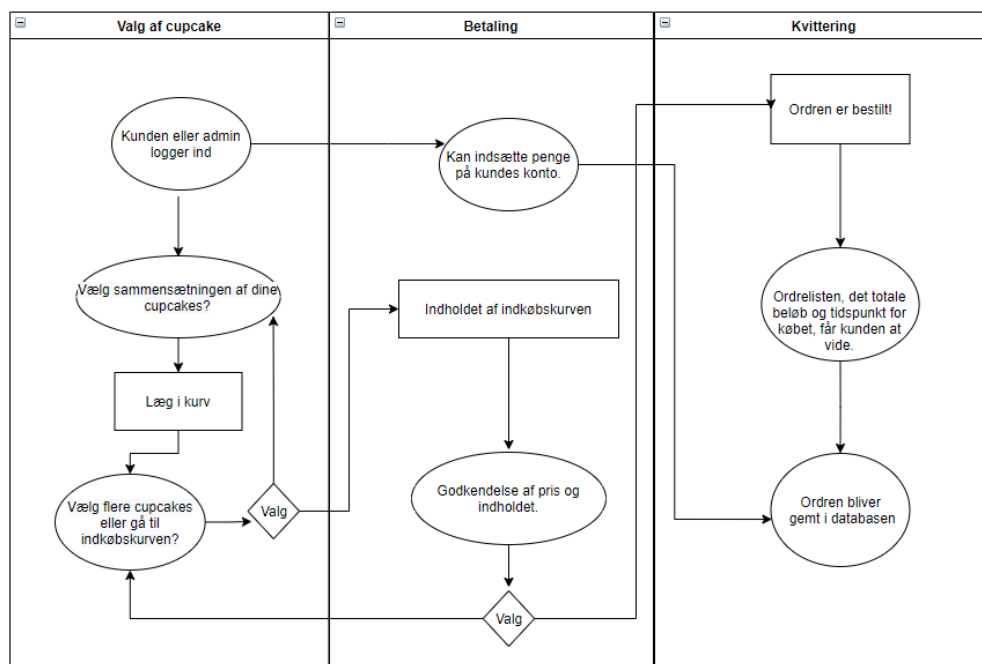




To be:

Brugeren logger ind på webshoppen, enten som admin eller kunde.

Kunden kan vælge, sin sammensætning af cupcakes og hvor mange, kunden kunne tænke sig. Derefter kan kunden lægge dem i kurven, hvor kunden så kan lægge flere cupcakes i kurven eller gå videre til indkøbskurven. Her kan kunden bekræfte ordren, kunden betaler for ordren og får en kvittering på bestillingen.



Særlige forhold

Informationerne som gemmes i session, er en brugers og administrators:

- email(brugeren/administratorens email)
- user(hvem er logget ind)
- userid(hvilket id har brugeren/administratoren)
- role(hvilken rolle er vedkommende: bruger/administrator)
- sum(prisen på bunde og toppings ganget med antallet)

Exceptions har vi benyttet til at få et overblik over eksempelvis if/else sætninger. Ved brug af vores exceptions, kan vi finde frem til mulige fejl og kontrollere de fejlmeddelelser, der vil komme når en exception bliver kastet. Vi har især taget brug af UserException.

Vi har valgt at udarbejde vores projekt med 2 forskellige brugertyper, customer og admin. De bruges på samme måde, men har adgang til forskellige ting. I forhold til JDBC er der ingen forskel.

Status for implementation

Vi er to i vores gruppe, så vi har haft en forhåbning om, at vi gerne ville kunne nå at lave de første seks user-stories. Vi har haft uventet problemer, i form af IntelliJ og MySQL gennem projektets forløb.

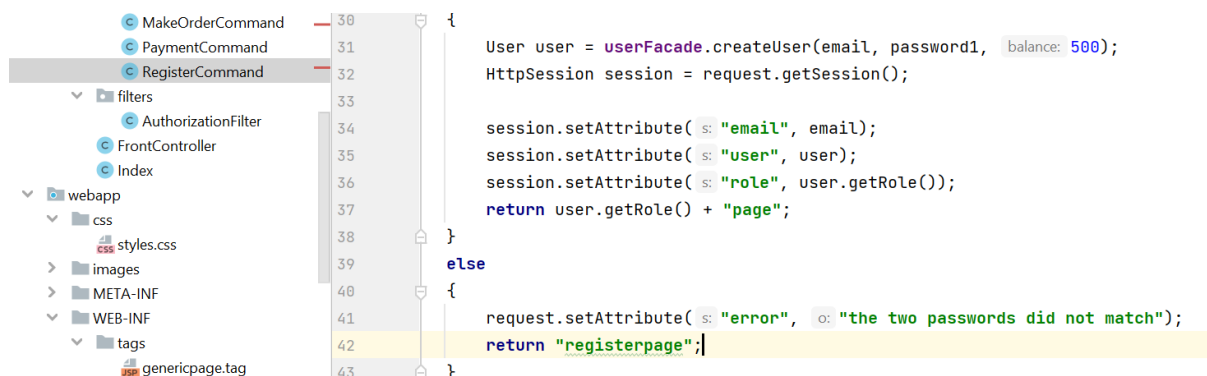
På jsp.siden ønskede vi, at selvom brugeren ikke er logget ind, at brugeren stadig ville kunne gå til betaling fra indkøbskurven. Det fik vi ikke sat op, så når brugeren er logget ind, går den tilbage til forsiden, hvor brugeren så må gå ind i ordren igen, for at kunne betale.

Vi kunne have brugt mere tid på styling og finjustering af hjemmesiden, hvilket kunne være med til, at gøre webshoppen mere brugervenlig.

Vi ville gerne, have arbejdet videre med afhentningen af cupcakesne. Kvitteringen måtte gerne have indeholdt, et tidspunkt på hvornår cupcakesne, ville være klar til afhentning.

Vi ville gerne, have haft udarbejdet selve admin siden bedre. Administratoren måtte gerne kunne tilgå, og have et bedre overblik over forskellige statistikker gennem hjemmesiden.

For at være kommet helt i mål, med de første seks user-stories, skulle vi have haft implementeret en form for betaling, når kunden bestiller ordren. Når kunden trykker bestil, i vores program, bliver der udskrevet en kvitteringen, som også bliver gemt i databasen. Der er gjort en funktion klar i RegisterCommand klassen, men vi nåede ikke at få funktionen implementeret færdig eller gjort det muligt gennem html.



Exceptions kunne vi godt have brugt mere tid på, da vi kunne lave mere brugervenlige beskeder, ved en fejl, så vi kunne forholde os bedre til det i koden.

Proces

Før projektstart, var vores mål at få en større forståelse af kodning, modeller og samarbejde. Vi ville forsøge vores bedste og lære af vores fejl.

Vi aftalte at afsætte hverdagene fra 9 til ca. 16 til projektopgaven. Det medvirkede til, at vi havde en klar aftale os selv imellem, hvor meget tid vi i hvert fald ville bruge på projektet. Vi har begge overholdt den aftale, der er blevet sat.

Under forløbet har vi haft en lærerig proces, vi har lært meget både af de ting der fungerede godt og af diverse fejl der opstod. Vi oplevede større problemer under forløbet, som gjorde vi blev nødt til at opsøge hjælp. Alt hjælp har været meget lærerig, da alle ikke tænker ens. Vi har fundet ud af, at det er vigtigt, at vi holder fast i hvad vi har gang i. For ikke at miste tråden, i for mange gode forslag.