MyCupCakeProject

Deltagere:

Anders Wade Jensen - cph-aj539@cphbusiness.dk - github: DarkDrinker

Nicolai Christian Dahl Pejtersen - cph-np158@cphbusiness.dk - github: PejterNico

Github link: https://github.com/DarkDrinker/Cupcakeproject_gruppeC

Youtube link: https://youtu.be/X-7IKqOB4LI

Klasse/semester: Datamatiker 2. Semester

Dato: 02.11.2023

Indholdsfortegnelse

Indledning

Dette projekt omhandler at cupcake forretningen Olskers, fra Bornholm, skulle have lavet en hjemmeside til deres fede forretning. Det er denne hjemmeside som vi, i dette projekt, har stået for at udarbejde. Projektet omhandlede at få udarbejdet en prototype hjemmeside, så Olskers kan se et udkast af hvordan deres fremtidige hjemmeside muligvis kunne komme til at se ud. Målgruppe for projektet er de kunder som handler hos Olskers cupcakes.

Baggrund

Den virksomhed som skal bruges systemet/websitet er Olskers Cupcakes, og det kan kunne benyttes både af Olskers ansatte som en administrator, men også af virksomhedens kunder som skal handle cupcakes hos Olskers.

Kunden, altså virksomheden Olskers, havde også nogle krav til hjemmesiden, som blandt andet omhandler at kunderne selv kan sammensætte deres cupcakes, ved at vælge mellem forskellige topping og bunde, og lægge dem i deres kurv og senere købe dem. Samtidig skulle kunderne kunne oprette sig som brugere og kunne logge ind på deres hjemmeside.. Derudover kunne Olskers også tænke sig at de som administratorer kunne se deres alle deres kunder samt kundernes ordre, og også kunne håndtere ordene hvis den ordre nu ikke blev til noget.

Teknologivalg

I projektet har vi brugt følgende teknologier til fremstilling af systemet:

- IntelliJ IDEA 2022.3.2
- Java 17
- Javalin 5.6.1
- Maven 4.0.0
- PostgreSQL
- HTML

Krav

Firmaets håb med dette system er at have en hjemmeside hvor kunderne kan oprette en profil, logge ind, og derefter være i stand til at kunne bestille cupcakes fra Olskers. Olskers håb er også, at hans ansatte eller hans "administratorer", som vi efterfølgende kalder “admin”,kan gå ind og se alle deres kunder samt kunders ordre. Derudover kunne de også godt tænke sig at admin har mulighed for at kunne rydde op i deres kundeforhold samt deres ordre så man ikke har gamle/fejlede ordrer unødigt liggende i systemet. Den værdi som systemet bidrager til Olskers virksomhed er at deres kunder har mulighed for at kunne bestille cupcakes hvor som helst fra og så senere have mulighed for at kunne afhente deres ordre i butikken.

Nedenfor er de User Stories som Olskers gerne kunne tænke sig implementeret i deres system:

US-1: Som kunde kan jeg bestille og betale cupcakes med en valgfri bund og top, så 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 Postgres, så en kunde kan betale for sine ordrer.

US-4: Som kunde kan jeg se mine valgte ordreline i en indkøbskurv, så jeg kan se den samlede pris.

US-5: Som kunde eller administrator kan jeg logge på systemet med email(username) 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.

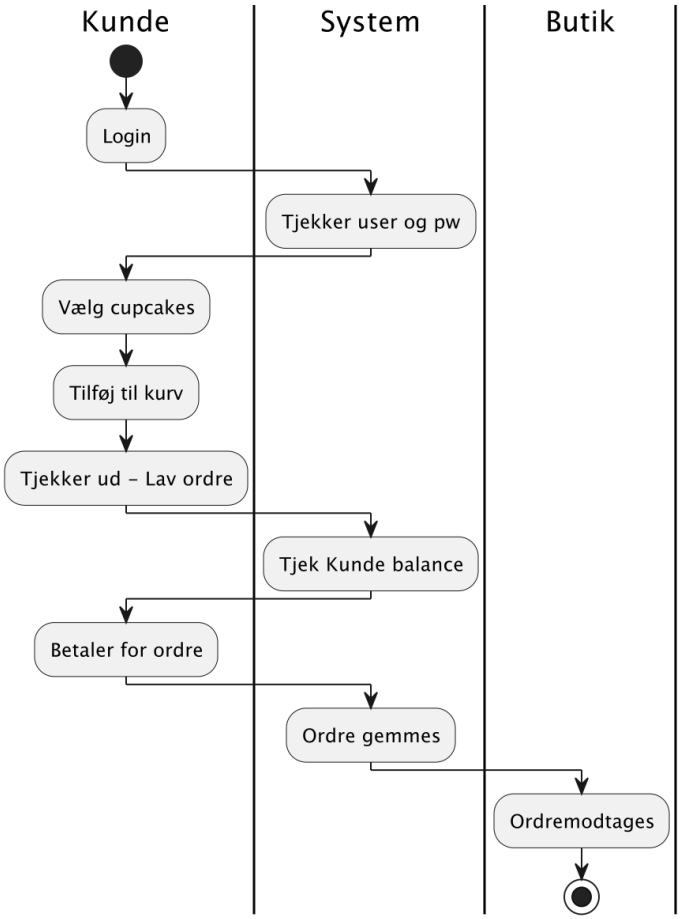
US-7: Som administrator kan jeg se alle kunder i systemet og deres ordrer, så jeg kan følge op på ordrer og holde styr på mine kunder.

US-8: Som kunde kan jeg fjerne en ordreline 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.

Aktivitetsdiagram

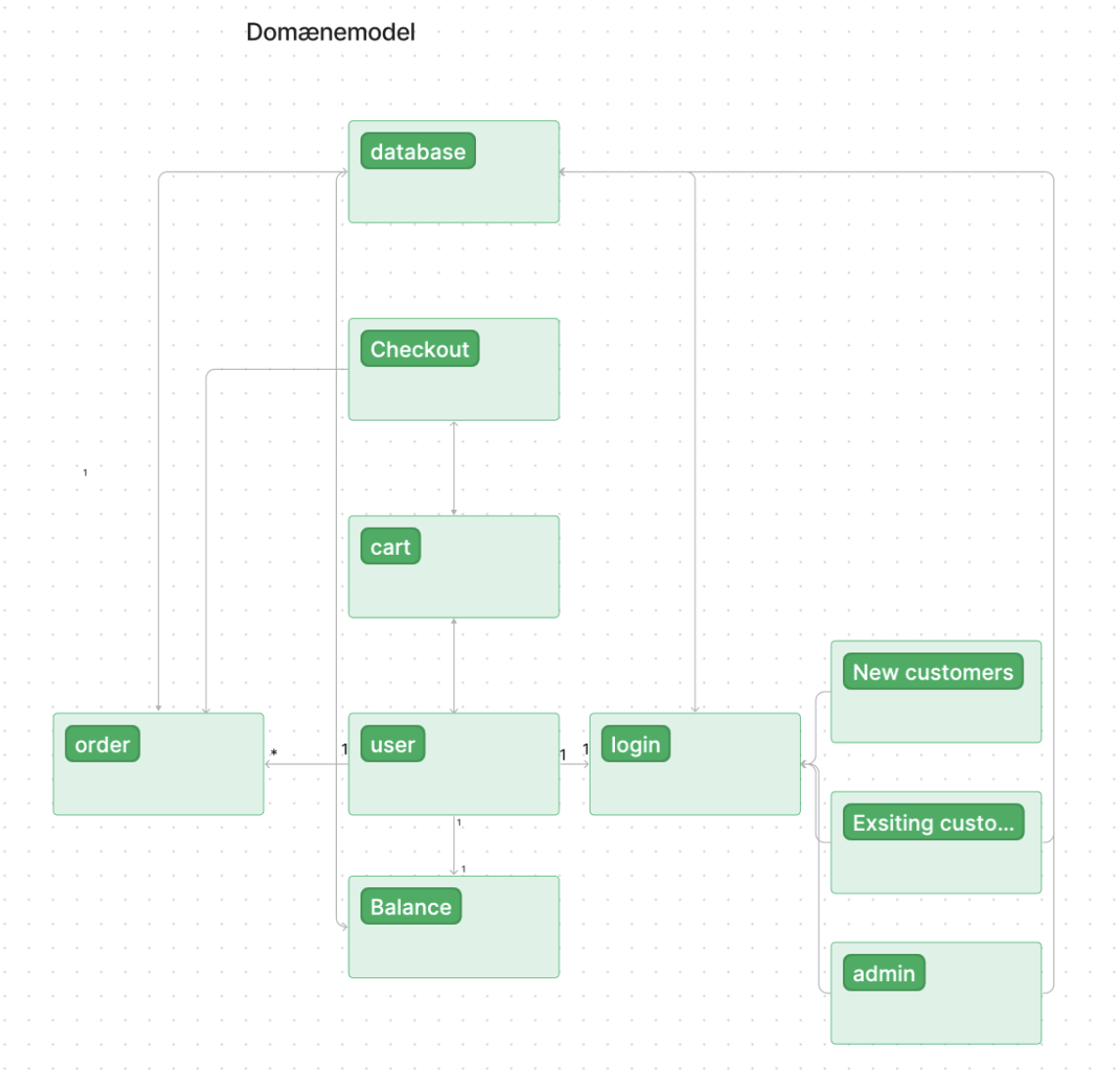
Aktivitetsdiagram for hvordan hjemmesiden kunne komme til at se ud i den endelige:



Domæne model og ER diagram

- Domænemodel:

Her nedenfor er domænemodellen over de forskellige elementer i systemet:

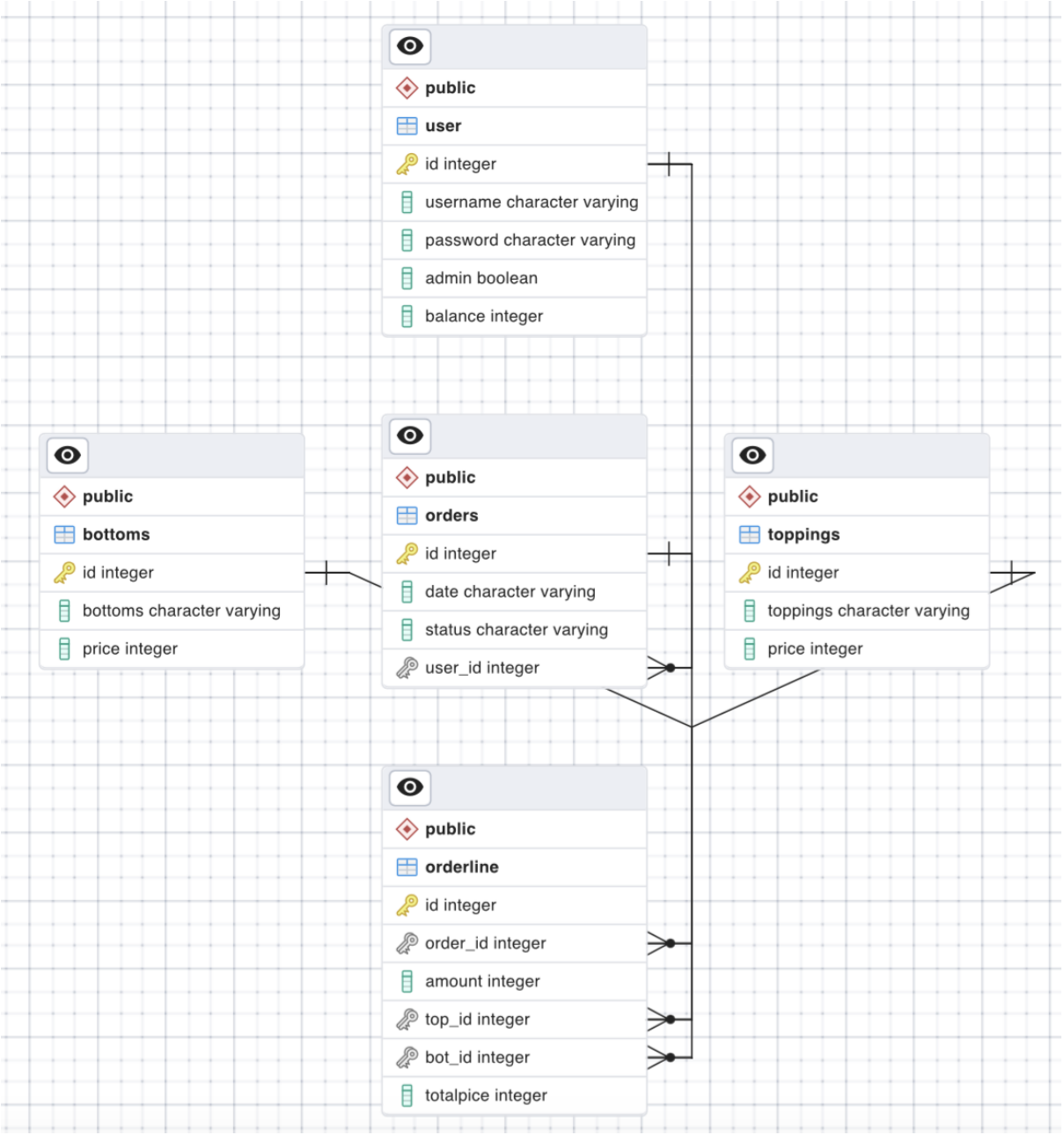


I systemet er der flere forskellige relationer, der er bl.a. en 1 til 1 relation mellem “user” og “login”. Det vil sige at en bruger kun kan have en bruger/login(Som er specificeret af username), det vil altså sige at en bruger kan i teorien godt have flere logins, men et username kan kun bruges en gang. Dermed at flere brugere ikke kan have samme username.

I systemet er der også en 1 til mange relation mellem “user” og brugeres “odrer”, det vil sige at en bruger sagtens kan have flere ordrer, altså have handlet over flere omgange.

- ER diagram

Her nedenfor er diagrammet over ERD, som beskriver relationerne i databasen:



I alle tabeller har vi anvendt en automatisk genereret ID. Alle de autogenererede ID'er fungerer som primærnøgle for de enkelte tabeller. Dette gør at vi ikke selv skal sende en ID med til databasen når vi opretter nye elementer, men at nye elementer selv får et tildelt.

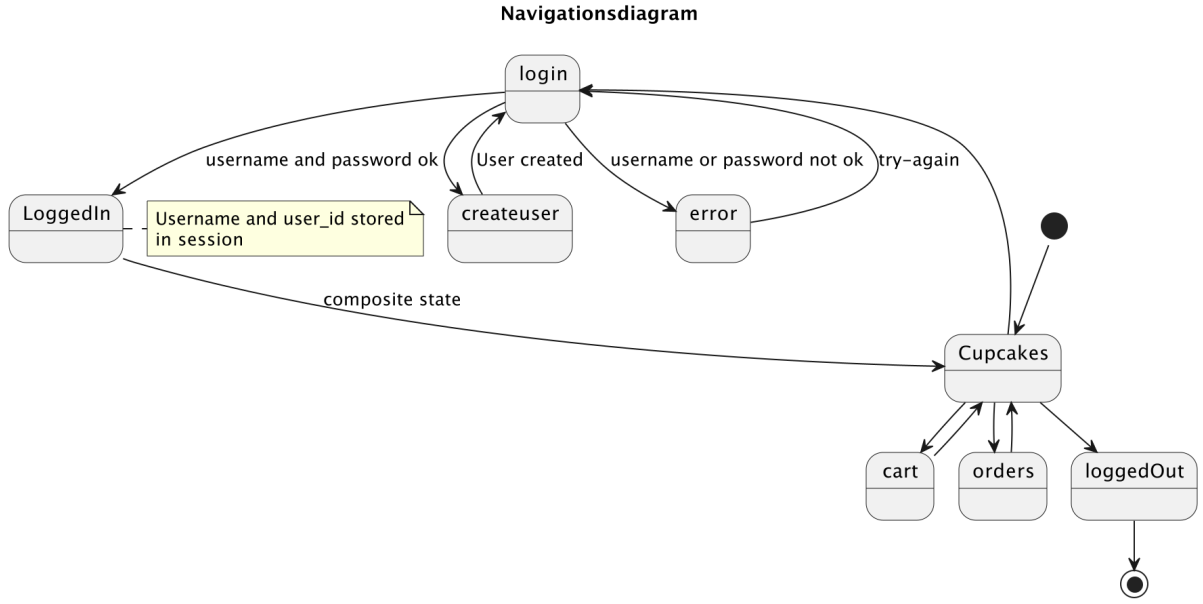
Derudover benytter vi os også af fremmednøgler i flere af tabellerne. I tabellen "orders" hvor vi benytter os af fremmed nøglen fra user i User_id, således at vi kan bruge user_Id til at finde den bruger, som er logget ind, ordre, så den pågældende bruger ikke får ordre fra andre brugere.

I tabellen orderline benytter vi os også af fremmednøgler fra bottom og topping, hvor at den benytter sig af Id fra bottom og Id fra topping til til at udvælge hvilken topping og bottom som kunden har lagt i "orderline"

Navigationsdiagram

På vores sider benytter vi os af en fælles navigation bar i toppen af alle sider, som gøre det muligt at kunne navigere gennem de forskellige sider, og dermed hoppe fra “cart” til “order” og tilbage til forsiden “cupcakes” igen.
På hjemmesiden har brugeren også mulighed for at kunne ligge cupcakes i deres cart, uden at være logget ind.
Siden “orders” kan til gengæld kun nås ved at man logger sig ind som bruger. Og skulle dermed have mulighed for at kunne se alle sine ordrer.
Fra login siden har man derudover også mulighed for at kunne oprette sig som en ny bruger, hvor man angiver sit username og password. Efter at man så har oprettet sig som bruger bliver man så ledt tilbage til login siden for derefter at logge ind.

Der var desværre udfordringer med at få UML til at sætte .jsp på så derfor kommer de nedenfor her:
Cupcakes.jsp
cart.jsp
orders.jsp
login.jsp
createuser.jsp



Særlige forhold

Her beskriver vi nogle af de særlige forhold vi har i projeket.

- Informationer der gemmes i session
    - Efter login gemmes: Navn(brugernavn), ID, Balance, isAdmin(admin status).
    - Vi havde også muligheden for at gemme ens kodeord, men vi tænker det er en god ide at lade vær.
- Den eneste exception vi håndterer er vores database exception.
    - Denne exception håndterer alle exceptions indblandet med forbindelsen, 'injection' af vores kode i DB.
    - Hvis vi var kommet videre i projektet, ville vi nok skulle bruge exceptions i forbindelsen og forsendelsen af data til backend og frontend.
- Vi har valgt at validere vores bruger login med sql.
    - `String sql = "select * from \"user\" where username=? and password=?"`
    - Herfra sætter vi bare det indtastede brugernavn og kode, vi har fået fra brugeren.
    - Alt efter om noget matcher eller ej, logger vi ind eller giver en fejlkode.
- Ift til sikkerhed når i logger ind. bruger vi prepared statements, så det ikke er muligt at lave SQL-injektioner.

Status på implementation

Det er dog ikke alle ting som vi har nået at få implementeret. Her nedenfor lister vi hvilke implementeringer vi har fået gennemført og hvilke som vi ikke har kunne nå at få implementeret:

Implementeringer som ikke er gennemført:

- Man kan ikke oprette sig som bruger.
- Orders er ikke særligt detaljerede. Kun de mest basale informationer.
- Du kan ikke købe ting i din kurv.
- Du kan ikke se en samlet pris for din kurv.
- Du kan ikke redigere eller slette enkelte ordrelinjer i din kurv.
- hvis du trykker på "home" fra din kurv, sletter den din nuværende kurv.
- Hvis du trykker på banneret, skulle den sende dig til forsiden. Det virker ikke.
- **Alle** funktioner indblandet med at du er admin/administrator virker ikke.

Implementeringer som fungere og kører på hjemmesiden:

- Brugeren kan logge ind.
- Brugeren kan logge ud.
- Vi viser den loggede ind i "nav-baren" på hver af vores hjemmesider.
- Efter at have logget ind, kan man se muligheden for "orders" og ikke før.
- Du kan lave dine egne cupcakes, som du har lyst.
- cupcakes der er ens med allerede lavet cupcakes i din kurv. vil bare tælle op i stedet.
- Du kan se beløbet på din nuværende ordrelinje, mens du laver din cupcake.
- Vores navigationsdiagram var simpelt til at starte med, men vi har nået alt derpå.
- Vi har opfyldt CRUD så længe det gav mening. fx UserController klassen, behøver ikke alle de funktioner for den opfylder sin funktion.

- Vi har brugt fragments til at nemt kunne beholde et ensartet look på vores hjemmesider. Dog har vi ikke brugt så lang tid på at style dem alle. Men vi prioriterede funktion over styling i denne omgang.

Proces

Vi havde i starten af projektet store forhåbninger til hvordan projektet skulle forløbe og hvilke forhåbninger vi havde i forhold til udarbejdningen af hjemmesiden. Vi brugte i starten god tid på at udarbejde modeller over hvordan vores tanker og visioner var for opbygningen af både databasen men også hvordan flowet på hjemmesiden skulle være.

Vi var dog kun 2 personer i gruppen, hvilket gjorde at vi desværre ikke nåede at få alt implementeret. Og derfor valgte vi også at fokusere på få de ting som fungere til at fungere, hvilket bla. er hvordan man bevæger sig rundt på hjemmesiden samt fokus på at hvordan man som bruger kunne logge ind og vælge sine cupcakes, med bund og top, og tilføje dem til sin kurv.

I forhold til uddelegeringen af arbejdet var det nemt at blive enige om, hvem der laver hvad og begyndte på de forskellige elementer i udarbejdelsen af hjemmesiden, derfor benyttede vi og ikke af “task board”, til at se hvad vi manglede og hvad vi skulle nå.

Dette har nok desværre også bidt os lidt bag i, da vi ikke havde en helt fast vision og ide om hvad der skulle laves og hvem der skulle stå for de enkelte elementer, og dermed havde vi heller ikke nogen deadlines for hvornår de enkelte elementer skulle være implementeret.

Derudover gik det også op for os hvor meget der også er i forhold til backend på hjemmesiden, hvor vi brugte meget tid på også at få layoutet på hjemmesiden til at se flot ud.

Til en anden gang vil vi nok også dele opgaven op i mindre delopgaver og mindre deadlines under forløbet og samtidig lave en mere fast plan over projektet og over de forskellige del elementer så vi nemmere kan holde styr og retning på projektet og de forskellige delelementer, også så vi mere ved hvor vi bla. skal sætte ind i forhold til en eventuel løsning på en af de udfordringer vi kunne have stødt på.