Vi valgte at arbejde videre med vores produktbacklog, så vi have noget mere at vise til vores præsentation men også for at se om der var nye relavante problemstillinger, som vi kunne tage op.  
  
Præsentation af produkt

1. Første use cases vi kiggede på var **”muligheden for at oprette en virksomhed”** - opretter en virksomhed, som vi vil bruge til at holde styr på hvilke teams de hver i sær har, så man ikke bare får vidst alle mulige team som ikke har noget at gøre med en brugers formål.
2. Create team – Her er der tilføjet en udvidelse, nemlig en **dropdown af companie**s, så det er selvfølgelige også blevet en krav til oprettelsesprocesses, som der også bliver tjekket om (så hvis man ikke angiver det, kommer der en fejlbesked
3. Til sidst har vi tilføjet en **Edit side** - hvor man har mulighed for se hvilke team der er i forskellige companies, men også muligheden for at redigere et team (tilføje og slette medlemmer.
   1. På nuværende tidspunkt er det dog ikke muligt at slette eller update teams (metoder er begge halvlavet, men vi skulle også i gang med fremlæggelse, så det må blive løst efter)

**Ustruktureret rest**

Vi opbygger hele tiden teknisk gæld, det ses også i vores REST og her har vi også set hvor stor betydning det kan have.

Det var faktisk først da vi gik i gang med vores Edit side, at vi stødte ind i problemet med en ustruktureret REST indtil videre, hvilke kom fordi vi fik 9 REST metoder mere, så vi nu har 18 forskellige metoder der kan kalde og som hedder noget forskelligt

Vi prøvede så vildt muligt at holde endpoint ens indenfor hver tabel. Så alle CRUD-metoderne til Member hed ”/member”, og det der så skilte dem ad, var selvfølge selve requesten (GET og POST) og de parametre man gav med.

Hvis billede af metoder der heller det samme.

Så tilføjede vi de nye metoder og så var der lige pludselig to GET-metoder inden for Member, de kan ikke hede det samme, det resulterede i at vi have en masse forskellige endpoint.

Hvis billede af medtoder der ikke hedder det samme

Så nu havde vi en masse metoder, der hed og gjorde noget forskelligt, så vi havde brug for at identificere

* Det var her vi begyndte at benytte **summary** funktion, her kunne vi nemlig beskriver hvad funktionen gjorde og side hvilken parametre man forventede at få med i kaldet.

Hvis billede af summary

Der var dog stadig mange scrolle op og ned, så vi valgte at indfører en indholdsfortegnelse i starten, så det gjorde det nemt ligt at finde metode.

Billede af indholdsfortegnelse

Fremadrettet – Skal indholdfortegnelse og summary laves sideløbende med interfacet og implementationen af interfacet.