

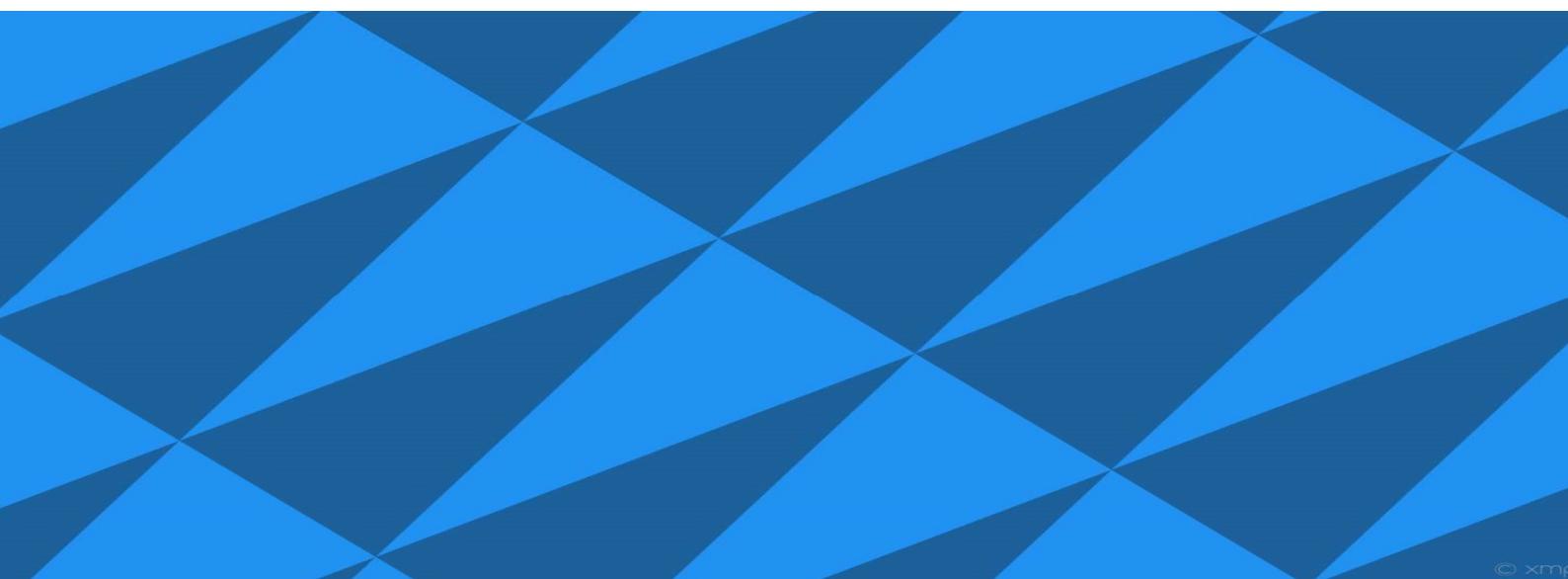
Zealand

HOVEDOPGAVE

Christoffer Dunk Ring, Ian Mañebo Frost,
Jonas Manley Pedersen & Kasper Kofoed

Vejleder: Vibeke Sandau

Produkt Link: [Git Labs](#)



© XM

ABSTRACT

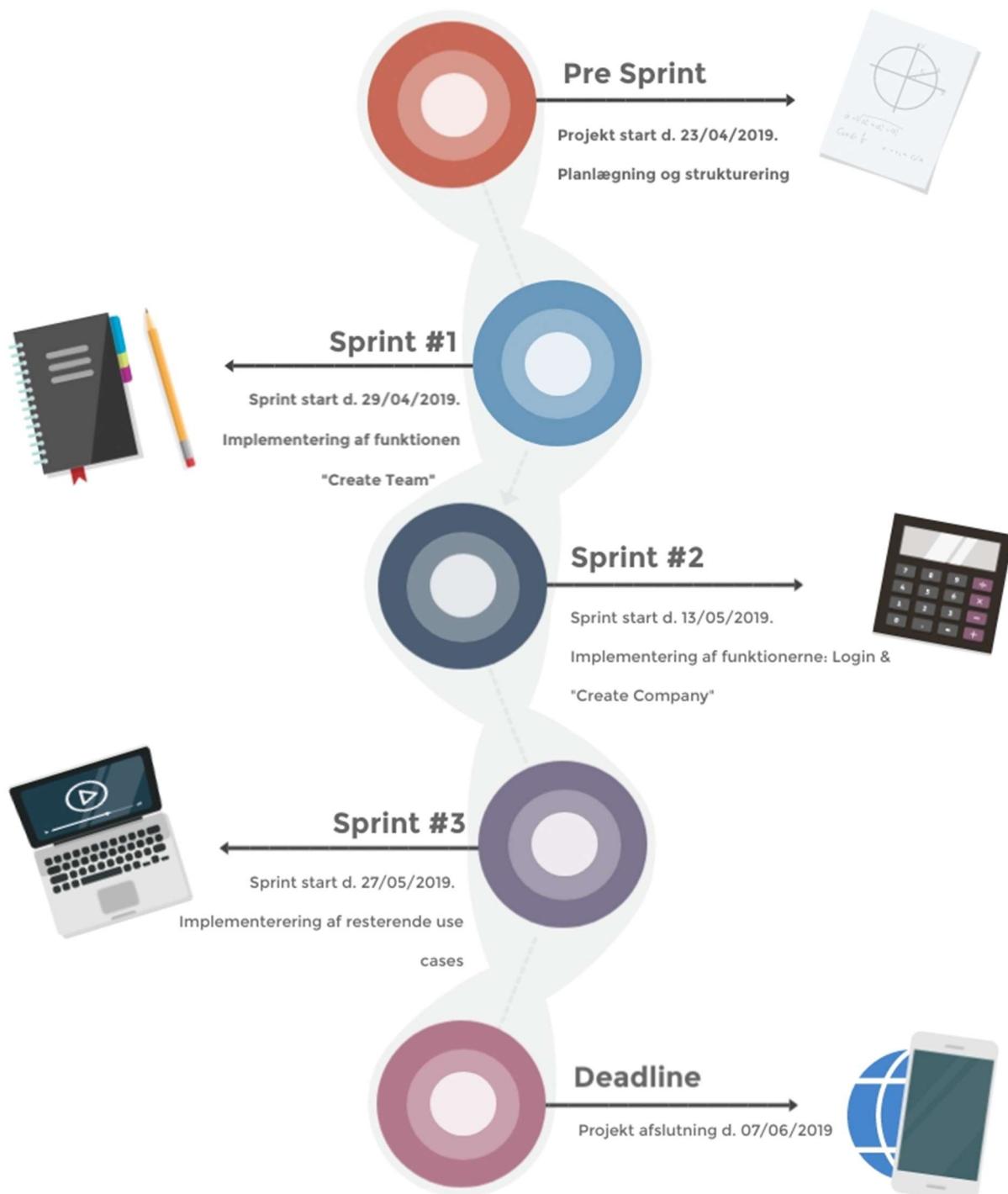
[Forfatter: Kasper, Christoffer]

The purpose of this final project paper serves to both give insight into how we developed as well as our thought process for the reader. How do you design and develop in a market that is already more or less saturated with personality tests and furthermore, do it for a customer residing in another country for free? Working with Business Mindset located in Northern Ireland, we quickly ran into communication problems - misunderstandings and a lack of response due to their busy schedules. Seeing as we're a side project for the owners we were understandably not their first priority. We had to apply our own ideas and thoughts to not get stuck and waste valuable time. Setting up our own databases and generating data to for us to use in our development.

We had to take a step back and take a look at the bigger picture and develop what we believed to be the best fitting solutions, given our customer's needs. We had to make the design by conscious reasoning, meaning that not only did we have to design the overall user experience from guess-work regarding what the customer might want, but the general structure of the project as well.

Overall experiencing firsthand what the tech industry has in store for us, has been an invaluable experience. The market isn't going to slow down, and neither should we, regardless of what obstacles we face. It is the adaptation to a given situation, to the best of our abilities that can really set us apart from our industry colleagues.

Projekt Timeline



Indhold

ABSTRACT	2
INDLEDNING	5
INTRODUKTION AF BUSINESS MINDSET	5
PROBLEMFORMULERING	7
RISIKOANALYSE.....	8
METODEVALG	10
KVALITETSSIKRING	15
ARKITEKTUR.....	16
FORRETNINGSANALYSE.....	17
Pre Sprint	21
INTRODUKTION.....	21
UDVIKLINGSMILJØ	23
ARBEJDSPROCES	25
DESIGN SPRINT	26
Sprint #1.....	28
SPRINT RETROSPECTIVE- PRE SPRINT	28
UDVÆLGELSE AF USER STORIES	30
KRAVÆNDRINGER D. 22/04-2019.....	32
SPRINT PLANNING.....	32
HVORDAN UDVIKLEDE VI	33
VERSIONSSTYRING	41
KRAVÆNDRINGER D. 29/04-2019.....	44
OPSUMMERING AF SPRINT #1	45
SPRINT REVIEW.....	47
Sprint #2.....	48
SPRINT RETROSPECTIVE	48
SPRINT PLANNING.....	50
HVORDAN UDVIKLEDE VI GUI DESIGN	52
REST METODER.....	54
JAVASCRIPT.....	55
CORS.....	56
FIREBASE & AUTHENTICATION	57
Selenium.....	61
INFORMATIVE WORKSPACE	63
OPSUMMERING AF SPRINT #2	64
SPRINT REVIEW.....	65
OBS! ÆNDRING AF PROBLEMFORMULERING	66
Sprint #3.....	67
SPRINT RETROSPECTIVE	67
HVORDAN UDVIKLEDE VI	69
TEST	72
PRODUCT REVIEW.....	73
SPRINT RETROSPECTIVE OVER ARBEJDSPROSESSEN.....	74
REFLEKSION	75
PERSONLIG REFLEKSION.....	76
KONKLUSION	78
LITTERATURLISTE	79
BILAG	81

INDLEDNING

[Forfatter: Ian, Jonas]

I forlængelse af vores praktikforløb, stiftede vi kendskab med det nordirske firma Business Mindset. På daværende tidspunkt, var deres arbejdsproces plaget af manuelt arbejde eftersom der som udgangspunkt først bliver sendt et invitationslink til personlighedstesten, dernæst skal testen tages af vedkommende. Business Mindset får disse testresultater til gennemgang samt, bearbejdelse. Efterfølgende vil test resultaterne blive evalueret ud fra en fysisk samtale.

På baggrund af alle samtalerne vil Business Mindset slutteligt være i stand til at sammensætte og vurdere hvor god teamets sammensætning af profiler er ift. at opnå succes er. (dettes gøres også manuelt).

Dette er en åbenlyst en længere trættende proces og derfor ønskede de at kunne automatisere dette.

Begge parter så således dette som en oplagt mulighed for både læring såvel som en mulighed for at kunne afprøve nogle nye ting i praksis. Dette sikrer dem en masse sparet ressourcer samt en mulighed for at få innovative løsninger. I forlængelse af automatiseringen af arbejdet, ønsker de at få implementeret en funktion der i stand til at danne et velstruktureret team, vurderet ud fra forskellige personligheds tendenser fundet ved hjælp af deres egen videnskabeligt understøttet personlighedstest for dem automatisk også.

Denne rapport dækker projektplanlægningen såvel som udviklingen af produktet hen over et 7 ugers langt forløb fra start til slut.

INTRODUKTION AF BUSINESS MINDSET

[Forfatter: Christoffer, Jonas]

Business Mindsets er en online platform hvor brugere kan tage en "personlighedstest" der kan fastslå om de er oplagte kandidater til enten at være entreprenører, entrepriser eller proces orienterede teammedlemmer.

Virksomheden består af følgende medlemmer:

- John - Business Mindset CEO og vores kunde.
- Andrew - Business Mindset CTO og vores IT-vejleder.
- Sorin - Server Admin
- 4 andre medarbejder vi ikke er blevet introduceret til

Testen viser hvor på en skala man ligger i forhold til disse tre kompetencer

- **Entreprenør:** En person der tager chancer for at organisere tilgængelige ressourcer for at starte sit eget firma
- **Enterprising:** En person med stærkt sammenlignelige kompetencer og kvaliteter som en entreprenør, men ikke ønsker at tage risikoen ved at starte eget firma
- **Proces orienteret:** Fungerer bedst ved at blive givet faste rammer for hvad der forventes af deres arbejde.



Business Mindsets vision er at give indsigt i et team gennem individet samt, at styrke teamets struktur gennem den enkelte persons kompetencer. Dette gøres ved at give dem indsigt i hvilken type person de er, så de ved hvilken rolle de bedst egner sig til. Dette gøres ift. entreprenør-ska-laen. (se ovenstående billede).

Hvorfor adskiller Business Mindset for andres personlighedstest?

Først og fremmest bygger testen på unikke data tilegnet fra den virkelige verden (*i form af en undersøgelse, hvor 50 store succesfulde entreprenører, er blevet testet i forhold til deres kendeteogn, holdninger og visioner*).

Derudover tages personlighedstesten med henblik på at sammensætte et unikt team som også adskiller sig fra andres personlighedstest da disse giver individet indblik i hvordan han/hun er som person. Med viden om hvordan bestemte personer agerer overfor hinanden i forhold til hvilken personlighedstype de er, gør at man kan strukturere teamet på en given måde, så der opnås bedst "flow".

Som bruger af Business Mindset har du muligheden for at identificere din personlighed spredt på 8 forskellige punkter:

- 4 forskellige motivationstyper - hvad der motiverer dig til at yde det bedste samt og hvordan det påvirker dine kollegaer både negativt og positivt
- Om man er konformist eller nonkonformist
- Tjekker om din personlighed består af Entreprenør-egenskab, Enterprising-egenskab eller procesorienteret opførelse.

Hvordan virker testen:

For at kunne deltage i testen er et invitationslink nødvendigt for at kunne registrere dig hos Business Mindset. Efter man er oprettet som medlem skal man svare på 42 spørgsmål som alle består af multiple choice. (Se billede til højre for eksempel)

The screenshot shows a digital questionnaire titled 'Questionnaire'. At the top right is a close button ('x'). Below the title is a horizontal bar with three colored segments: light blue, orange, and green. The main area contains a numbered question and two options for each. Question 1 asks: 'People tend to have a preference for one statement or the other. Please click on which statement best represents your approach.' It lists two statements:

- Before I start a task I prefer to be given clear instructions about what I am expected to do.
- Before I start a task I prefer to be given a general idea and then get on with it.

Når alle spørgsmål er blevet besvaret, bliver man videreført til resultatsiden.

Resultat af test:

Resultatsiden indeholder en oversigt over hvordan dine scorer er over de 8 forskellige kategorier

- **Entrepreneurial Succes Factor** - En overordnet score for hvor store chancer du som individ har for at få succes som entreprenør.
- **Broad Minded** - Man ser det fra det store billede først og arbejder sig ind ad, fremfor at have stort fokus på detaljerne.
- **Intuitiveness** - Man er god til at tænke i andre baner.
- **Non-conforming** - Om man er omstillingsparat og klar på nye og udfordrende opgaver.
- **Boundary Developer** - Nedbryder gamle traditionelle metoder, ved at tilegne sig ny viden, skaber en boundary developer nye metoder som er bedre end de traditionelle.
- **Boundary Keeper** - Man føler sig tryg inden for kendte rammer og har intet begær for at sætte spørgsmålstegn ved disse rammer.
- **Boundary Pusher** - Elsker at søge nye udfordringer og skubbe trygge rammer på alle tænkelige måder. de bryder sig ikke om at følge helt konkrete retningslinjer.

- **Boundary Filler** - Eftersom boundary developer og pusher ikke er modstridende der kan føre til stridigheder og ineffektivitet, er det oftest nødvendigt at inddrage en boundary filler som fungerer som mellemled, som kan ”oversætte” for part til part.

Scoren ud fra hver kategori udregnes ud fra en bestemt algoritme, baseret på vægten af de besvarede spørgsmål. Kategorierne indeholder en skala der strækker sig fra 1-9, hvor 9 betyder man er meget karakteristisk indenfor kategorien - dette betyder at hvis man har en lav score, så er man det modsatte. F.eks. hvis man har en lav score i Boundary Pusher er man meget plandrevet.

PROBLEMFORMULERING

[Forfatter: Kasper, Ian, Jonas & Christoffer]

Problemformuleringen er et eller flere konkrete spørgsmål af et problem, som skal undersøges. Vores problemformulering er dog anderledes på det punkt, at Business Mindset har meldt ud, at kravene til funktionalitet kan ændre sig drastisk, når de får mere indsigt i deres ønsket produkt. Derfor er vores problemformulering en ”*løbende problemformulering*” der højst sandsynligt vil ændre sig gennem forløbet.

- **Hvordan udvikler vi en ny funktion, der på baggrund af en personlighedstest kan identificere om et team er ideelt for at opnå succes.**
 - Hvordan kan man styre en udviklingsproces når man udvikler for en kunde bosat i et andet land
 - Hvordan imødekommer man en kundes vision der ikke er konkret nok til at fastslå hvad deres krav er

RISIKOANALYSE

[Forfatter: Kasper, Jonas & Ian]

For at kunne imødekomme eventuelle problemer vi måtte kunne støde på. Bliver vi først nødt til at overveje hvilke problemer vi anser for at være gældende i vores situation. Vi er ved at udvikle et produkt med et nordisk firma og har på pågældende tidspunkt under 7 uger tilbage indtil deadline. Dette er hvad vi anser for at være gældende problemstillinger:

- **Lav**
- **Mellem**
- **Høj**

Risk 01 | Kundekontakt

Beskrivelse:

Det er problematisk at vi på et givent tidspunkt ikke bare kan komme i direkte kontakt med kunden. Grundnet distancen mellem os, men også dels fordi at de selv har mange bolde i luften på én gang.

Påvirkning:

Kan gøre at teamet i længere perioder står uden arbejde eller med spørgsmål der skal besvares før at process kan fortsætte

Ansvarlig:

Teamlederen

Mindskning af Risici:

Teamet er nødsaget til at overveje om der skal arbejdes på andre use cases og fortsat informere kunden.

Risk 02 | Sygdom

Beskrivelse:

Sygdom anser vi i forårsmånedene stadig for at være sig en meget gældende problemstilling da temperaturen veksler op og ned hele tiden.

Påvirkning:

Hvis et teammedlem oplever sygdom kan han ikke bidrage med input eller produktivitet til projektet deraf mistet arbejdskraft

Ansvarlig:

Teamlederen: Håndtering af situationen med et medlem mindre.

Medlem: Skal indberette sygdom

Mindskning af Risici:

Afhængigt af hvor slem sygdommen er, vil vedkommende enten fortsat arbejde på deres tasks ellers videregives de til en anden i teamet.

Risk 03 | Research

Beskrivelse:

Eftersom projektet indeholder ny viden og teori, gør at vi bliver nødt til at sætte os ind i det nye stof. Som kan være mere eller mindre tidskrævende

Påvirkning:

Kan fjerne fokus for vigtige opgaver og proces for at læse om ligegyldige teori

Ansvarlig:

Teamlederen: Hjælp med at finde mere relevant læsestof

Mindskning af Risici:

Evaluere jævnligt og spørg fagpersoner til råds

Risk 04 | Diskussion

Beskrivelse:

Vi har mange forskellige holdninger til arbejdet og hvordan det skal gøres. Derfor ender vi tit med at diskutere diverse muligheder, og til tider kommer dette til at tage mere tid end hvad vi ønsker.

Påvirkning:

Bruger utrolig mange kræfter og energi på nedprioriteret ting og kan være med til at skabe splid internt.

Ansvarlig:

Ordstyrer: Hvis ordstyreren føler diskussionen har taget en drejning som ikke vedrører det relevante emne, skal vedkommende bryde ind.

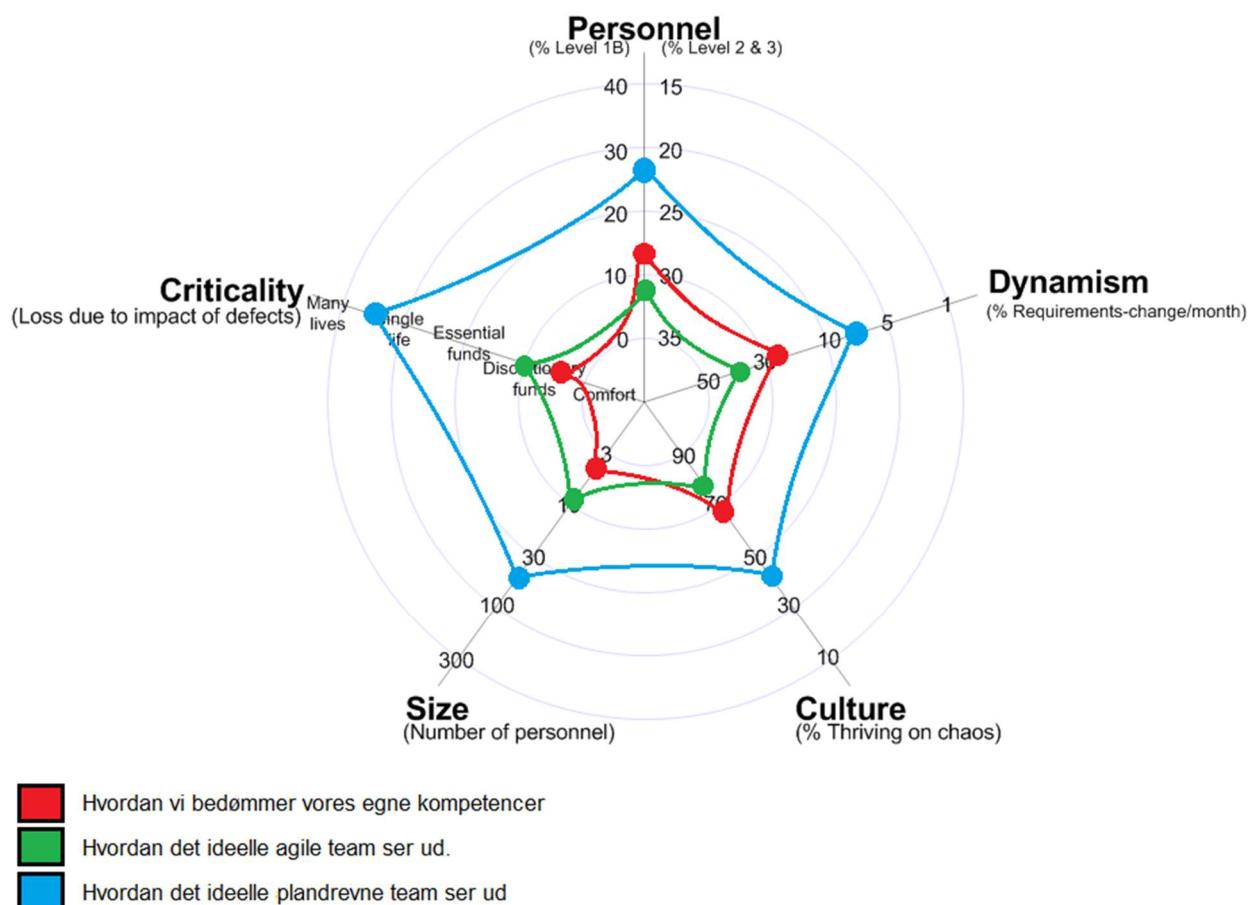
Mindskning af Risici:

Teamet må afsætte maks. 20 min. til at færdiggøre diskussionen. I tilfælde af intern splid tager Ordstyreren og de vedrørende en samtale med henblik på at afklare konflikten.

METODEVALG

[Forfatter: Kasper, Jonas, Ian & Christoffer]

I forlængelse af at vi indledningsvis har fokus på planlægning af projektet, er det også nødvendigt at vi lægger et fokus på vores valg af metoder. Netop for at sikre den, højest mulige effektivitet i forløbet, så diverse risici så vidt muligt undgås og vi når i mål til deadline. Med erfaring fra tidligere projekter, har vi kunne drage nytte af værktøjer vi også anvendte dengang, f.eks. Boehm & Turner modellen. Hvilket er et teoretisk værktøj og formålet ved brugen af Boehm & Turner modellen, er essentielt set at give overblik over teamets kompetencer, derved giver det os en bedre idé om hvordan vi i teamet arbejder bedst, om det er agilt uden faste rammer eller mere plandrevet med fastsatte rammer.



Det smarte ved modellen er som sagt at den giver os et visuelt overblik over vores kompetencer. (*Ud fra billedet ovenfor*) ses det tydeligt som beskrevet, at vi bedømmer vores egne kompetencer som værende mere oplagte i brug med en agil tilgang frem for en plandrevne tilgang.

Personnel: er brugt som et estimat til at komme frem til teamets arbejdsniveau. Måden hvorpå man gør dette, er ved at kigge på de enkelte niveauer man ifølge modellen kan være og derved se på de kriterier man skal opfylde for at opnå disse. I dette projekt har vi alle studeret i ca. 2.5 år og har været gennem erhvervspraktik. Vi har arbejdet med både agile såvel som plandrevne arbejdsmetoder og kan tilpasse disse nogenlunde efter behov. Vi kan skrive velstruktureret kode og har lettere ved at danne os et overblik i projektet end førhen, grundet vores erfaring. Derfor vurderer vi

os selv, niveaumæssigt, at ligge omkring **Niveau 2**. Hvilket ifølge modellen er et ideelt niveau for at kunne arbejde med den agile tilgang.

Niveau
-1
Kan have teknisk kunnen, men er ikke i stand eller villig til at samarbejde eller følge delte metoder
1B
Har de mest basale færdigheder og kan skrive simpel kode, basal refactoring samt simpel testning. Kan følge mere simple arbejdsmetoder
1A
Har god fundamental viden og kan skrive mere avanceret kode, god refactoring, mere kompleks testning. Kan benytte sig af mere avancerede arbejdsmetoder
2
Kan tilpasse diverse arbejdsmetoder og tilrettelægge dem efter behov
3
Kan helt ændre på strukturen af en arbejdsmetode og derved bryde metodens reglement for at tilpasse den til en given situation

Dynamism: er hvor vi kigger nærmere på hvor dynamisk anlagte vi er. Dette vurderer vi ud fra hvor gode vi er til at håndtere kravændringer. I og med at vores kunde er bosat i Nordjylland og vi ikke nødvendigvis har mulighed for at være i hyppig kontakt, anser vi muligheden for at kravændringer kan opstå oftere frem for sjældnere da kunden har længere tid mellem møder til at gøre sig nye overvejelser. Vi har i teamet samarbejdet gennem hele uddannelsesforløbet, og kender derfor til hinandens kompetencer, og vi vurderer blandt andet derfor at vi har et godt grundlag for at være gode til at håndtere en masse ændringer undervejs i projektet. En høj dynamism er oplagt til agilitet, der netop har sit fundament i at acceptere kravændringer undervejs i processen.

Culture: er hvor vi vurderer hvor omstillingsparate vi er ift. prioritering af arbejdsopgaver. For at uddybe yderligere, er culture punktet hvor vi kigger på hvordan vi foretrækker at arbejde. Kan man bedst lide at arbejde med faste rammer. Eller foretrækker man mere frihed i sit arbejde samt muligheden for at træffe flere beslutninger? Vi mener selv vi bedst arbejder med muligheden for at træffe nogle valg undervejs frem for at nødvendigvis at "låse" os fast i en fastlagt plan fra start.

Size: does matter, større teams på f.eks. 30 medlemmer, kræver langt mere planlægning end et team på vores størrelse, blot 4. Vi har muligheden for at arbejde mere hurtigt og dynamisk og har mindre behov for faste rammer. Det er let for os at koordinere medlemmer og vi kan let danne et overblik over hinandens arbejdsopgaver.

Criticality: hvor store konsekvenser har det hvis vores produkt skulle ende med at være fejlagtig. Med andre ord, hvor kritisk vores software er for firmaet. Tager vi udgangspunkt i udviklingen af f.eks. en netbank, er det af højeste prioritet at softwaren bare virker som den skal uden undtagelser. Hvorimod at vi anser at en nedbrudt personlighedstest højest vil ende med at koste firmaet en smule penge i trafik indtil den, er oppe igen.

Altså kan vi ud fra denne model lettere argumentere for vores valg af den agile tilgang. Nemlig at vi i bund og grund egner os bedst til at arbejde med valgfrihed såvel som et ønske om ikke at låse os selv fast i en fast plan fra start.

Grundet forløbslængden på en cirka 7 ugers tid, vil det i vores tilfælde ikke give mening at gøre brug af SCRUMS standard længde sprints som er på en 3-4 ugers, hvilket ville resultere i at vi maks ville nå at få feedback på 2 iterationer. Men derimod tilpasser det til at være sprint længder på 2 ugers tid. Dette er fordelen ved SCRUMs versatilitet, nemlig evnen til ikke at være fastlåst i én enkelt måde at skulle gøre det hele på. Der er selvfølgelig fordele og ulemper ved begge, da man med henholdsvis meget korte iterationer, ender man med at have mindre tid til at finpudse produktet, dog kan vi i sidste ende gøre op for dette ved langt mere regelmæssig feedback. Skulle vi have valgt at arbejde med 3-4 ugers iterationer, er det i bund og grund en modsætning i den forstand at vi nu har længere tid til at refactor og generelt set sikre et langt mere kvalitetssikret produkt men vi kan ende med at have uforudsete kravændringer vi ikke har tid til inden for deadline. Det hele bunder ud i at det som udgangspunkt er en god ting at have en veldefineret i iterations længde i vores projekt, en varighed der ikke varierer hen over forløbet. Dette hjælper os med at holde arbejdsrytmen kørende og giver os et bedre overblik over hvornår vi skal nå hvad og hvorfor.

På baggrund af tidligere erfaring har vi udvalgt en række aktiviteter, som vi mener er med til at hjælpe og strukturere vores projekt optimalt.

SCRUM

SCRUM er en agil udviklingsmetode, som benyttes med henblik på at kunne skabe software på en mere dynamisk og fleksibel måde. Et hovedprincip ved SCRUM er at det er en arbejds metode, der lægger meget stor vægt på at kunden undervejs i arbejdsprocessen kan ændre mening og dermed ændre kravene uden at det nødvendigvis skal skabe problemer for teamet, hvor man i modsætning ved brug af eksempelvis vandfaltsmodellen slet ikke ville kunne få dette til at lade sig gøre uden kritiske konsekvenser for udviklingen af produktet. Med kortere såkaldte sprints er det langt lettere at kunne imødekomme og endda forvente ændringer under projektet. Da der altid er mulighed for at prioritere opgaverne når de er delt op i mindre bidder.

Sprints

Et sprint i projektet vil være af en varighed på 2 uger, som gør os i stand til at have 3 sprints i alt. Vores sprints består af flere ting: Sprint planning, sprint retrospective, sprint review, daily scrum m.m.

Sprint hjælper os med at arbejde iterativt, agilt og derved får vi feedback løbende fra kunden.

Daily SCRUM

Daily scrum vil blive afholdt kl. 08:00 hver dag.

Teamet samles om en tavle og laver en opsamling hvor der planlægges dagens mål, recap og hver enkelte arbejdsopgaver.

Daily scrum hjælper os med at skabe et overblik over hvad der skal laves og om vi har travlt.

Sprint Planning

Sprint Planning udføres i starten af hvert sprint, ud fra product backlog medtages så mange use cases som teamet har estimeret de kan nå per sprint.

Ved at planlægge vores sprint sikre, at vi hele tiden ved hvad vi skal nå, hvor vi skal hen og hvad vi mangler.

Sprint Retrospective

I slutningen af hvert sprint vil teamet i fællesskab udfører denne aktivitet.

Ved at reflektere over tidligere sprints, om hvad der har været godt eller skidt, kan vi forbedre det næste sprint. Dette er med til at skabe værdi for teamet da vi ønsker at opnå det mest produktive sprint.

Sprint Review

I slutningen af hvert sprint vil alt kørende "software" blive udleveret til kunden. Så teamet kan få evalueret produktet og få eventuelle rettelser til krav eller produkt.

Design Sprint

Design sprint består fem faser

- **Understand** - Er fasen hvor vi klarlægger kundens krav og hvilke der er vigtigst.
- **Diverge** - Her udformer vi hver især vores ide til hvordan en potentiel prototype skal se ud, i forhold til de use cases vi fandt frem til i "Understand" fasen
- **Decide** - Formålet med denne fase er at beslutte hvad en ideal prototype skal indeholde. Dette gøres ved at vi hver især fremviser vores udkast og derefter diskutere, hvilket design vores fælles prototype skal indeholde.
- **Prototype** - Der udarbejdes en prototype pga. af fasen understand, hvor vi får kendskab til kundens krav.
- **Validate** - I denne sidste fase bliver prototypen valideret af kunden, som giver os tilbagemeldung på vores prototype og giver nye krav hvis, den ikke er tilfredsstillende. Dermed kører processen i cyklus indtil kunden accepterer prototypen, som den afgørende.

Design sprintet har til formål at få teamet til at forstå kravene hurtigere og nemmere, samt at få det bekræftet tidligt i processen.

Vi benytter os af Design Sprint i pre-sprintet.

Design sprint blev anbefalet af Andrew, da han mere eller mindre bruger det i starten af alle hans projekter.

Product Backlog

Vi vil benytte os af Trello, da det er et enkelt og let værktøj at bruge.

Product backloggen vil blive oprettet som det første efter estimeringen af vores use cases (tasks). Backlogen giver teamet en liste med arbejdsopgaver. Vores product backlog indeholder følgende:

- Navnet på given arbejdsopgave
- Kort beskrivelse af given arbejdsopgave
- Hvor vigtig den enkelte arbejdsopgave er
- Hvor lang tid det estimeres at arbejdsopgaven tager

Estimering

For at estimere vores use cases benytter vi os af planning poker (se afsnittet om planning poker).

Planning poker vil blive benyttet i starten af sprintet når størstedelen af kravene er indsamlet.

Vi har valgt at estimere alle vores use cases (tasks), som gør os i stand til at finde vores teams velocity, så vi fremadrettet kan vurdere hurtigere om vi kan have færre eller flere use cases (tasks) med.

Derudover bruges estimeringens pointene, som en metrik til vores Burndown Charts over sprintet og projektet.

Use Cases

Use cases er letforståelige scenarier der viser hvordan systemet skal interagere uden benyttelse af tekniske begreber.

Use cases bliver skrevet så straks kravene er forstået.

Vi benytter os af use cases for bedre at kunne forstå vores arbejdsopgaver samt, sikre vi får alle de vigtige elementer med i produktet.

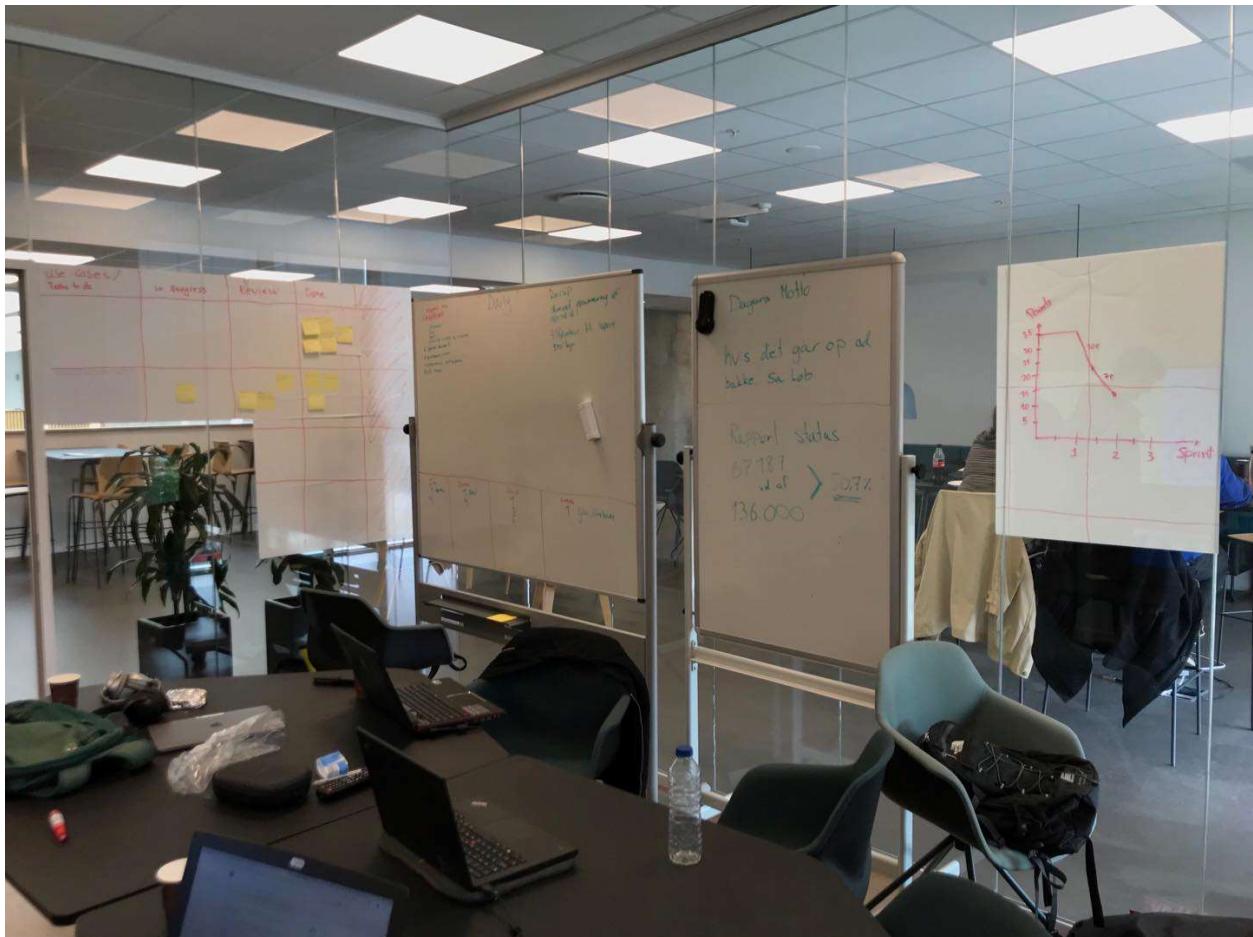
Burndown Chart

Før daily scrum vil scrum master opdatere alle burndown charts og gøre klar til at give teamet et hurtigt overblik, netop over hvordan projektet skrider frem og om vi kan nå at færdiggøre alle planlagte arbejdsopgaver til tiden.

Informative Workspace

Vi benytter os af tavler, sticky notes og whiteboard papir til at dokumentere alt der giver processen værdi - det informative workspace vil blive opdateret under daily scrum.

Med et informative workspace kan man hurtigt lave ændringer, få overblik, skitsere tanker og gøre det så alle kan se det.



De følgende elementer indgår i vores informative workspace:

- **Dagens Motto**
 - Vi har valgt at have en dagens motto, som skrives af en udvalgt person (valgt af scrum-masteren) til at komme op med et kreativt eller nytænklede motto, der skal motivere teamet til at komme igennem dagen. (det kan også gøres i form af en joke)
 - Det skal ikke tage mere end 2 min.
- **Daily scrum tavle**
 - **Recap** - Her opsummeres hvilke arbejdsopgaver der blev færdiggjort fra dagen før - det giver nemlig god mening at starte der hvor man sluttede.
I tilfælde af et ikke fremmødt medlem, kan det også bruges til en kort opsummering for vedkommende.
 - **Dagens mål** - som er helt konkrete arbejdsopgaver som teamet mener at de kan færdiggøre til slutningen af dagen. Ved at krydse små færdig opgave af, motivere virkelig meget i forhold til fx. at sidde med noget i 4 uger og ikke rigtig føle at man kommer nogle veje.
 - **Arbejdsfordeling** - hver arbejdsopgave i dagens mål, vil blive tildelt et tal 1...2...3... hvert tal vil så blive skrevet nedenfor det medlem som har ansvaret for at klare

opgaven - det går både så hvert medlem ved hvad han skal arbejde på, men også så scrum-masteren nemmere kan danne et overblik over hvad alle i teamet laver.

- **Burndown chart**
 - Er medtaget for at skabe transparency i form af hvor langt vi er nået i processen, hvad vi har tilbage og hvor lang tid vi har til at færdiggøre det.

KVALITETSSIKRING

[Forfatter: Ian]

For at sikre det produkt vi udvikler består af høj kvalitet og uden mangler har vi valgt at indføre forskellige testmetoder under vores projektforløb. De følgende testmetoder hjælper os med at imødekomme vores interne krav om at lave et velfungerende software program:

Sit together:

Ved at vi i teamet møder op på skolen hver dag og arbejder i fællesrum, er det nemmere at kommunikere samt bidrage med inputs til eventuelle tvivl omkring arbejdsopgaver, dette gør vi mindsker misforståelser i teamet.

Ved at vende emnet i gruppen bliver det set fra forskellige synsvinkler samt man ikke skal vente længere tid på svar, end hvis samtalen tog sted over besked, må man antage at problemet bliver løst hurtigere.

Acceptance test:

I hvert sprint review leverer vi det software, vi har udviklet i indeværende sprint, til kunden for at vi sikre at projektet har rette kurs mod målet. Acceptance tests er med til at skabe overensstemmelse blandt alle inkluderet i projektet.

Hver mandag har teamets product-owneren et såkaldt "accept-brief". Proceduren er at der i tidsrummet 7:30-8:30, vil der blive afsat 15-45 min til at have en dialog med kunden, hvor product-owner kan fortælle hvad teamet i øjeblikket har fokus på, hvordan det går og hvad de næste dage byder på - Grunden er, at vi på den måde hele tiden er i tæt kontakt med kunden, så krav ændring, udskydning af arbejde m.m. kan imødekommes.

Det er bl.a. her kunden kan komme med ændringer, hvis der skulle være kommet noget siden sidst.

Ved at have denne tætte kundekontakt, gør at vi i teamet spilder mindst muligt tid i forhold til evt. kravændringer og teamet bliver 100% transparent overfor kunden, som gør at kunden hele tiden er med på hvad der sker og i sidste ende ikke står som et spørgsmålstejn, for nu har de været med i hele processen.

Softwaretest

Eftersom vi ikke har et program til at teste vores kode automatisk, tjekker vi hver funktion selv. Ved at benytte os af Postman tester vi vores HTTP call requests fra vores API, for at tjekke på forhånd om den vil fungere i programmet, dette sparer os for tid og gør koden pålidelig. Når vi opdager kode fejlen tidligt i forløbet er der meget mindre der skal ændres på, end hvis den var implementeret i produktet.

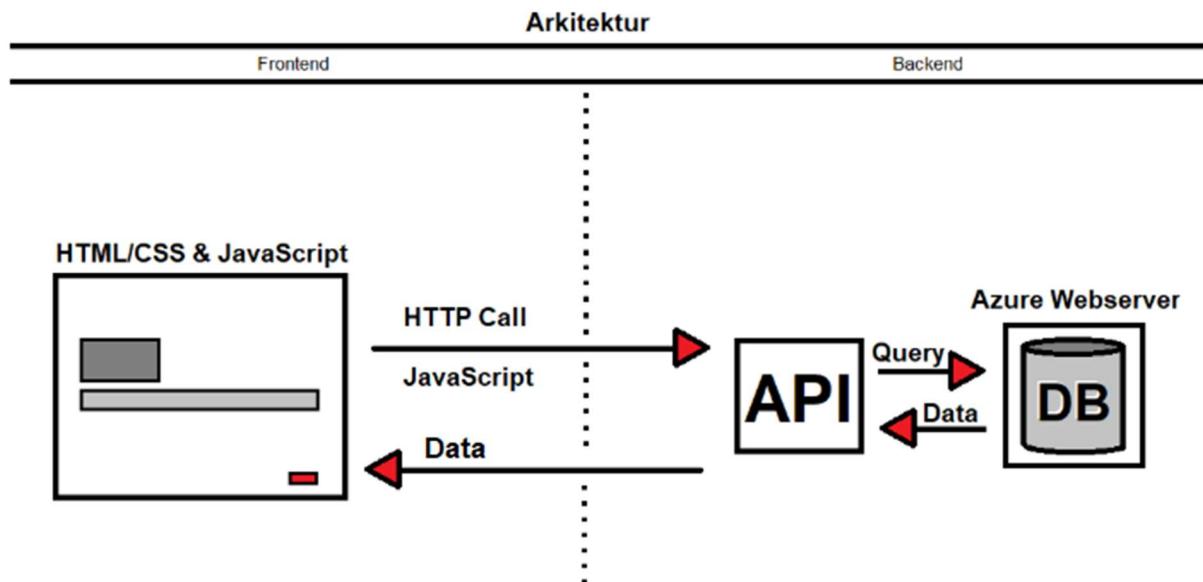
GUI test selenium

Selenium er en webdriver man benytter sig af når man vil lave GUI funktionelle tests. Vi ønsker at skabe en automatiseret test script i Visual Studio som kan køre på vores hjemmeside, så vi slipper

for at tjekke manuelt alle funktioner af hver dag. Ved at køre regression test på vores produkt sikrer vi også at produktet formår at bibeholde den høje kvalitet.

ARKITEKTUR

[Forfatter: Christoffer, Jonas]



Ovenstående er vores nuværende opsættelse af vores produkt arkitektur.

1. I frontend delen afventer vi brugerinput (klik og tekst).
2. Ved brugerinput, sendes der en efterspørgsel på et stykke data (HTTP call) til vores API. Dette kan være et bestemt firma der skal bruges til en vilkårlig opgave.
3. Vores API sender denne query videre til vores web server (Database).
4. Web Serveren sender den efterspurgte data tilbage til brugeren som i frontend delen vil få det visualiseret.

FORRETNINGSANALYSE

[Forfatter: Christoffer, Ian Jonas & Kasper]

SWOT

Som vi førhen har nævnt i vores aktivitets tabel, benyttede vi os af forretningsmodellen SWOT, som har til formål at sikre os at vi på forhånd ved at vi har en plads på markedet med vores produkt.

SWOT
<p>Strengths</p> <ul style="list-style-type: none">• Stærkt videnskabeligt understøttet personlighedstest baseret på mange års forskning af menneskehjernen• Kan vise dig hvilke roller dit team mangler for at blive en solid enhed og giver indsigt i hver individs rolle i et team• Kan vurdere hvordan dit team er struktureret ift. til at opnå succes• Automatisere manuelt arbejde og streamliner en nuværende længere og besværlig proces
<p>Weaknesses</p> <ul style="list-style-type: none">• Mange personlighedstest har fået et dårligt ry. Dette skyldes en tilgang der skal være let og hurtig, folk gider ikke bruge meget tid. Derfor er sider som 16 personalities, en smule misvisende, da alle mennesker er koges ned til kun 16 personligheder og folk sættes derfor lidt i båse.• Videnskaben bag ved en væsentlig mængde personlighedstest er ikke fagligt anerkendte. Dette medfører at folk ender med ikke at stole på produktet og ignorerer det
<p>Opportunities</p> <ul style="list-style-type: none">• Med nye perspektiver indenfor forskningen af menneskehjernen, har produktet potentiale til på længere sigt at styrke arbejdsmarkedet gennem langt flere velstruktureret teams og dermed skabe flere jobåbninger i fremtiden.
<p>Threats</p> <ul style="list-style-type: none">• Skulle det imod al forventning vise sig at teorien bagved vores produkt skulle blive modbevist, er dette en meget uheldig position for firmaet.• Der findes allerede mange store veletableret konkurrenter på markedet med lignende produkter. Det kan være svært at skille sig ud fra mængden

Porter 5 Forces

Når nu vi allerede har benyttet os af SWOT, vælger vi stadig at benytte os af endnu en forretningsanalyse. Forskellene mellem disse kan være svært at skelne mellem. Man kan se på SWOT, som værende dit firmas velvære i industrien. Hvorimod at Porter 5 Forces(**P5F**), kan anses som at være industriens overordnede velvære. Så samlet set, ser vi både på om vores firma kan være konkurrencedygtige, såvel som vi kigger på om det overhovedet er værd for os at træde ind på markedet. Dette er til fordel for os, da vi ønsker på bedst mulig vis at optimere vores produkt ift. kundens behov såvel som markedets behov.

Porter's 5 Forces
Threat of new entrants (LOW) <ul style="list-style-type: none">• Omkostningsfuldt at tilslutte sig industrien• Kræver meget forskning for at adskille sig fra konkurrenter
Power of supplier (HIGH) <ul style="list-style-type: none">• Business Mindset er selv supplier af deres data da deres produkt er baseret på deres egen forskning
Competitive rivalry (HIGH) <ul style="list-style-type: none">• Fuldkommen konkurrence (mange konkurrenter)
Power of buyer (HIGH) <ul style="list-style-type: none">• Produkterne er ens i hvad de tilbyder• Moderat mængde produkter at vælge mellem
Threat of substitution (MEDIUM) <ul style="list-style-type: none">• Allerede veletableret personlighedstests• Professionelle tests, psykologer, HR

Business Model Canvas

Vi har udarbejdet en canvas analyse af projektet for at sikre os selv endnu mere overblik over processen, netop for at overskueliggøre hele forløbet endnu mere. Det er et beskrivende indblik i vores prioriteringer med henblik på at uddybe hvad vi mener der skal til for os for at kunne nå i mål i sidste ende.

Business Model Canvas	
Key Partners	<ul style="list-style-type: none">• Vibeke Sandau<ul style="list-style-type: none">• Hovedopgave vejleder, hjælper med feedback og idéer• Business Mindset<ul style="list-style-type: none">• Vores kunde, kommer med krav til produktet
Key Activities	<ul style="list-style-type: none">• Kravindsamling• Planlægning, strukturering• Adgang til Business Mindset serveren• Research af emneområde• Feedback
Key Resources	<ul style="list-style-type: none">• Opholdsrum<ul style="list-style-type: none">• Plads til informative workspace• Fagperson<ul style="list-style-type: none">• Relevant viden vedr. hjernevidenskab• Relevant viden vedr. system strukturering samt, generel viden inden for IT
Value Propositions	<ul style="list-style-type: none">• Innovation<ul style="list-style-type: none">• Vi giver nyt syn på et allerede eksisterende produkt• Indsigt<ul style="list-style-type: none">• Vi giver teamledere mulighed for at kunne få indsigt i deres teams samt, få et visuelt overblik med feedback• Vi giver teamledere mulighed for at kunne få indsigt i deres teams samt, få et visuelt feedback
Customer Relationships	<ul style="list-style-type: none">• Self Service<ul style="list-style-type: none">• Indebærer at kunden/teamlederen i fremtiden skal kunne udføre processen selv• Personal Assistance<ul style="list-style-type: none">• Indebærer at der er høj kundekontakt
Customer Segments	<ul style="list-style-type: none">• Business Mindset Administration• Teamledere

Forretningsanalyser bunder ud i problemløsning. Måden hvorpå vi har benyttet SWOT og P5F, er at vi til dels har opsat SWOT på funktionen vi udvikler fremfor selve firmaet og P5F på selve markedet og firmaet. Dette opdeler analyserne tilstrækkeligt nok til at vi ikke ender med 2 resultater der er meget ens da de begge eventuelt var blevet lavet på firmaet.

En stor kontrast mellem de to analyser er desuden at, SWOT er meget individuelt for firmaet og kræver flere interne overvejelser end P5F, hvilket automatisk resulterer i et meget mere unikt resultat der formentlig ikke ligner andre firmaers. En P5F analyse derimod vil være meget ens henover industriens firmaer, på trods af dette er det dog stadig en god måde at få dannet sig et overblik over hvad det er der skal til for at kunne holde sig konkurrencedygtig ift. de andre konkurrenter på markedet.

Da vi som tredjepart i dette scenarie, i bund og grund ikke er nødsaget til at lave forretningsanalyser for firmaet. Har vi dog stadig valgt at benytte os af værktøjerne, grundet at vi ser det som en værdifuld måde for os at kunne få udnyttet vores krav fra kunden yderligere samt for at kunne se mere kritisk på vores arbejde og få udnyttet styrkerne og få optimeret funktionen de rette steder.

Pre Sprint

INTRODUKTION

[Forfatter: Jonas]

Inden projektets reelle start besluttede vi at sætte én uge af til et pre sprint.

I pre sprintet ville vi:

- Strukturere teamet i form af hvilke forskellige roller hvert medlem skulle have ud igennem hele projektet, så vi dermed hver især kendte vores ansvarsområde, som gør at der ikke er en person som skal huske det hele - *vi uddeler ansvaret*.
- *Beslutte vores værktøjer samt udviklingsmiljøet*.
- *Derudover er det også i pre sprintet vi fastlægger vores arbejdsproces, så når projektet starter, ved vi nøjagtigt hvad vi skal beskæftige fra første sekundt*.
- *Til sidst vil vi også bruge tiden på at forstå kundens krav - som vi gør vha. et design sprint*.

Vi har også valgt at lave en disponering af sider i vores rapport, så vi har en god ide om hvor meget hvert afsnits skal fylde.

Siderne er i lavet som et interval, hvor det mindste er minimumskravet og største er maksimum kravet.

Estimering
Forside
• 1 Side
Indholdsfortegnelse
• 1-2 Sider
Indledning
• 1-2 Sider
Problemformulering
• 1 Side
Hovedafsnit
• 50 - 74 Sider
Refleksion
• 2-3 Sider
Konklusion
• 1 Side

Roller

Først og fremmest skulle vi blive enige om hvilke roller der gav mening at have med i projektet og hvad de udvalgte roller så indebar.

Når det var på plads diskuterede vi hver person i teamets styrker, svagheder og kompetencer og ved så at holde disse op imod hver rolle, kunne vi nå frem til det bedste match.

Roller
Scrum Master <i>Christoffer Dunk Ring</i> Scrum Masteren har til ansvar at sørge for Scrum processen bliver gennemført.
Logbog <i>Ian Mañebo Frost</i> Logbog holderen står for at dokumentere hver dag i projekts forløb. Dette skaber værdi for holdet da man senere hen i forløbet kan finde præcis hvad der er blevet foretaget på en givne dato.
Litteratur <i>Hele Teamet</i> Der skal holdes orden på anvendt litteratur på ethvert givent tidspunkt under projektet.
Product Owner <i>Jonas Manley Pedersen</i> Product owneren er den eneste person, der er ansvarlig for håndtering af produktets backlog og skal forstå kravene bedre end teamet (som fungere som kunde, når kunden ikke er tilstede). Derudover har product owneren ansvaret for at føre ordet under skype møder og generelt håndtere alt kontakt med kunden.
Udvikling <i>Hele Teamet</i> Som developers samarbejder vi alle om at nå frem til kundens ønskede produkt.
Korrektur <i>Kasper Kofoed</i> Korrekturlæseren sørger for at få læst alt arbejde igennem for stavfejl mm. For at sikre en høj kvalitet i afleveret materiale.
Design <i>Kasper Kofoed</i> Som grafiker står du for design og det generelle udseende af både produkt såvel som rapport.
Ordstyrer <i>Ian Mañebo Frost</i> Som ordstyrer, har man ansvar for at lede diskussionen den rigtige vej og stoppe samtaler der ikke har nogen som helst reference til kerneproblemets, som startede diskussionen i første omgang.

UDVIKLINGSMILJØ

[Forfatter: Kasper, Ian, Jonas & Christoffer]

Med udviklingsmiljø menes der hvilke værktøjer, udviklingssprog og IDE vi vil benytte os af for at sætte rammerne op for vores projekt - mange af tingene vil sættes op før sprint #1, så teamet netop ikke skal bruge tid og kræfter på dette, hvor arbejdsindsatsen burde ligge på at udvikle.

Når udviklingsmiljøet er sat op giver det os et solidt fundament når første sprint skydes i gang - vi ved hvor vi gemmer vores ting (google docs) hvordan vi styrer vores projekt (trello) og hvilke sprog vi har tænkt at benytte os af (HTML, CSS, C# & JavaScript).

Vigtigere er at vi har gjort os overvejelserne og derved begrundet vores valg.

De værktøjer og redskaber vi har valgt at inddrage i vores udviklingsmiljø, er medtaget på baggrund af tidligere erfaringer, anbefalinger, ressourcer og krav fra kunden.

Miljø
<p>Trello <i>Projektplanlægning</i> Hvorfor: Ved brug af Trello har vi mulighed for at skabe vores product backlog. Vores product backlog skaber overblik over de arbejdsopgaver som teamet er i gang med, hvilke de mangler og hvilke de er færdige med. Arbejdsopgaver er udelukket i forhold til programmering.</p> <p>Grundlag for valg: Tidligere erfaring og anbefalinger fra Christoffers chef, Cormac.</p>
<p>Google Documents <i>Rapport, Dokumentation</i> Hvorfor: For at alle i teamet kan skrive på rapporten samtidig har vi oprettet et online Google Docs.</p> <p>Grundlag for valg: Tidligere erfaring, den bedste måde i vores optik at skrive samtidig i ét dokument. (<i>det med ikke at kunne skrive samtidig eller have 4 forskellige dokumenter, er ALT for sjusket og ustruktureret</i>)</p>
<p>Visual Studio Code <i>Integrated Development Environment</i> Hvorfor: Her kodes der i Javascript for at tilknytte vores visuelle hjemmeside med funktionerne fra vores REST-API. Vi</p> <p>Grundlag for valg: Valgt baseret på det tekniske aspekt og tidligere erfaring. (<i>Vi kunne f.eks. ikke få Atom til at køre en live server, men havde ingen problemer med at sætte det op i Visual Studio Code også er der mere hjælp ift. syntax fejl i Visual studio code kontra Atom</i>)</p>

Git

Versionsstyring

Hvorfor:

Vi har valgt at bruge git som versionskontrol program, som vi både bruger til at opbevare vores projekt filer, men også til at review hinandens kode. For ved brug af git kan vi nemt lave flere branches, hvor vi kan fordele færdig kode og kode der skal kontrolleres, dette bliver fint visualiseret gennem Gitlab's hjemmeside.

Grundlag for valg:

Erhvervserfaring baseret på Christoffers praktikforløb i Belfast - Fungerer bedre end GitHub Desktop da der er flere muligheder og commands der kan udføres, dog er der ikke et visuelt interface at interagere med. Git Bash består af **Git** (som er et versionsstyringssystem og **Bash** som er et *unix shell command line interface*)

One Drive

Opbevarelse af filer

Hvorfor:

Ved at benytte os af Onedrive har vi muligheden for at uploadere relevante dokumenter som alle i teamet har adgang til. Herunder har vi vores logbog, e-mails og andre relevante dokumenter ift projektet. Eftersom alle har adgang til dokumenterne kan vores logbog, burndown-chart og andre aktiviteter altid blive udfyldt, hvis der opstår sygdom.

Grundlag for valg:

Da skolen stiller onedrive til rådighed gratis for studerende og funktionalitet er meget sammenligning med dropbox og andre produkter, fandt vi det oplagt at gå med det vi får stillet til rådighed.

Azure

Database

Hvorfor:

Azure er et hosting website, som vi vil bruger til at deploy og opbevare vores database samt REST services, så de er online og derved let tilgængelige.

Grundlag for valg:

Tidligere erfaring - Alt database arbejde vi har lavet under vores studieforløb har været i Azure - Vi har også kigget på mongodb, men har ikke så meget erfaring..

Open Broadcaster Software

Optagelser af møder

Hvorfor:

OBS optager alt hvad der foregår på skærmen både lyd og billeder. For ikke at overlade vigtige samtaler til hukommelse har vi valgt at optage alle vores skype møder med OBS, så vi altid kan gengive klippet, hvis kravene står sløjt i hukommelsen. Dette øger chancerne for at skabe et produkt vores kunde er tilfredse med.

Grundlag for valg:

Anbefaling - Andrew anbefalede os at bruge OBS til at optage vigtige skype møder, så man altid har muligheden for at gense mødet som en flue på væggen. (Hvis man er i tvivl om en detalje)

HTML, CSS, JavaScript, C#

Kodesprog

Hvorfor:

HTML, CSS og Javascript er de sprog vi har mest erfaring i, når det kommer til at kode hjemmesider.

C# er det sprog vi har kendskab til i forhold til udviklingen af REST og database.

Grundlag for valg:

Efter samtalen med Andrew i Belfast, forklarede han den nuværende hjemmeside er kodet i Laravel og PHP, men anbefalede os at kode vores extension fra bunden i et sprog vi havde mere erfaring med (ellers gik der for meget tid på research af framework og alt det eksisterende)

Selenium

Test

Hvorfor:

Selenium er et webbaseret testing værktøj som automatiserer og foreviger brugertest, som vil køres når der er lavet ændringer på hjemmesiden på baggrund af et script.

Grundlag for valg:

Ian har under hans praktikforløb i Belfast stiftet bekendtskab med Selenium og set hvordan man i virksomheder benytter sig af automatiserede tests for at spare tid. Vi har valgt at indføre dette så vores applikation altid vil blive testet igennem for vores funktioner og GUI virker som de skal.

ARBEJDSPROCES

[Forfatter: Jonas, Christoffer, Ian & Kasper]

At have en arbejdsproces før projektets start, er som tidligere nævnt for at vi slipper for at bruge en masse tid i det første spring til at finde ud af hvad vi egentlig skal lave.

Sprint dag #1

Sprint Retrospective

- Fælles afstemning på hvad der virker, hvad der skal stoppes og hvad der skal forbedres.

Kravindsamling

- Møder med kunde via skype og e-mail korrencespondancer.

Planlægning & Estimering

- Planning poker. (*med use cases*)
- Prioritising product backlog. (*med kunde*)

Sprint Planning

- Hvilke use cases skal medtages i sprintet.
- Når en use case lægges i sprint backloggen oprettes der tasks til denne.

Hverdage i sprint

Daily Scrum

- Først klarlægges dagens mål dernæst recap af dagen forinden og til sidst uddelegeres tasks/arbejdsopgaver til gruppemedlemmer og skrives op på en tavle.
- Updater informative workspace.

Konkret Arbejde

- Med dette punkt menes tid afsat til udførelse af de konkrete arbejdsopgaver planlagt i forbindelse med daily scrum.

Opsamling

- Alle i teamet samles og vender tanker om dagens produktivitet og konkrete arbejde. (*Samt skriver noter til daily scrum for næste dag*)

Logbog

- I slutningen af hver arbejdsdag skriver logbog holderen et kort referat af hele dagen. (Logbogen vil blive benyttet som et løbende redskab til at kigge tilbage i tiden)

Feedback

- Vi vil løbende skrive mails om krav, mock-ups eller spørgsmål som vi ønsker feedback på. Det vil ikke sker hver dag, men fylder en stor del af hverdagene.

Slutningen af hver uge

Feedback

- Sidste hverdag i hver uge modtager vi HOP-vejleder til samtale/statusmøde om hvordan projektet skrider frem og kommer med feedback til rapporten

Slutningen af sprint

Sprint Review

- Feedback fra kunden og acceptance test

DESIGN SPRINT

[Forfatter: Christoffer, Jonas & Kasper]

Understand

Som tidligere nævnt i aktivitets-afsnittet er design sprintet fasen hvor vi prøver at forstå kundens krav. Dette gjorde vi ved hjælp af et skype møde, hvor Jon og Andrew præsenterede deres vision omkring produktet og hvad de helt konkret ville have os til at udvikle.

På baggrund af mødet havde vi en god forståelse for produktet og indsamlet de krav som vi skulle beskæftige os med:

As an administrator:

I want to see a dashboard, to quickly get an overview of the composition in the team So that i have insight in the team

I want visual feedback that lets me interpret the test results more easily so I have a better overview

I want to be able to differentiate various elements of the site through color coding so its faster to notice the different elements

I want to read an in-depth report for the team, covering their test results so I have a deeper knowledge

I want to be able to see the team members personality tendencies, so i can better guide them into their roles.

I want to be able to see a more insightful description of the individual members of the team, upon clicking on them so that i have an overview of the team

I want to be able to create a team so i can see team insight

I want to be able to add new members to an existing team so i can fulfill already existing teams

I want to be able to delete team members so that non-relevant members can be deleted from the team

I want to be able to delete teams so that wrongfully made team can be deleted

Diverge

Ved at følge diverge fasen udarbejdede de fremmødte i teamet hver deres prototype.

Der blev valgt at teamet kun arbejdede med de krav der omhandlede dashboardet og dens funktioner, da vi på daværende tidspunkt ikke havde fået noget information om hvad det indebar at oprette et team (det skulle business mindset lige afklare internt).

Efter hvert medlem havde udarbejde en prototype, skulle alle medlemmer fremlægge denne ved at beskrive den visuelt og dens funktionalitet.

De resterende medlemmer af teamet gav deres feedback og kritik af den fremlæggende prototype. Dette gav en bedre forståelse for den enkeltes prototype - når alle havde været gennemgået, gik teamet videre til anden fase.

Decide

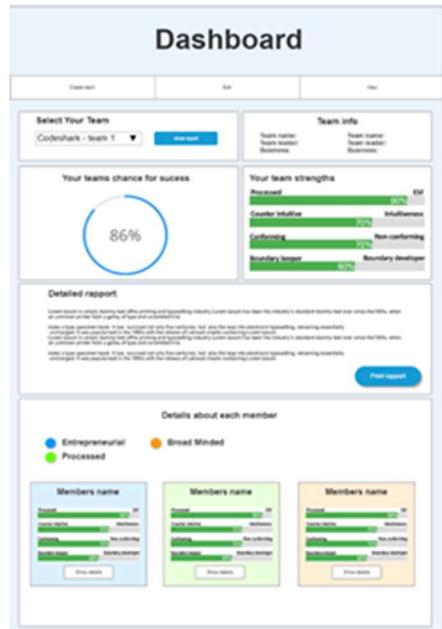
Vi kom frem til de elementer fra hver vores prototype vi syntes bedst om, som vi hver især også syntes var de mest væsentlige for at opfylde kundens krav. Vi lærte undervejs at det var vigtigt ikke at fastlåse sig på sin egen prototype, men derimod forsøge at skabe konsensus og gå på kompromis mellem flere prototyper.

Prototype

I forlængelsen af decide-fasen, udarbejdes der en konkret prototype med alle de elementer som teamet har fastlagt sig på, prototypen kan ses til højre:

Validate

Prototypen blev sendt til kunden så John & Andrew kunne se om vores udkast var noget der matchede det de søgte, og vigtigst af alt om vi havde forstået kravene og var på rette vej. (*tilbagemeldingen skete først om lørdagen, og derfor er resterende tekst medtaget i første sprint*)



Sprint #1

SPRINT RETROSPECTIVE- PRE SPRINT

[Forfatter: Ian]

Fredag d. 26/04-2019 efter vores møde med Vibeke, satte vi os ned og diskuterede hvad vi havde brug for af ændringer for at imødekomme vores mål ift. følge den tidsplan vi havde lagt. Vi har valgt at dele retrospective op i tre emner som henholdsvis står for hvad vi har gjort godt, hvad vi skal gøre bedre og hvad vi ikke får noget udbytte ud af og derfor ikke fortsætte med at gøre.

Keep:

Daily scrum:

Daily scrum har været udført hver morgen for at klarlægge hvad dagens arbejdsopgave har været, det har givet et stort udbytte til teamet da medlemmer altid visuelt kan se på tavlen hvilke arbejdsopgaver man skal arbejde på, hvis man bliver færdig med en task. Udeover at planlægge dagens arbejdsopgaver benytter vi også af recap deraf får vi også reflekteret over hvad vi brugte dagen forinden på.

Ugentligt møde med HOP-vejleder:

Hver fredag eftermiddag holder vi et møde med vores HOP-vejleder Vibeke for at få feedback på ugens arbejde på rapporten. Vi har oftest haft tvivl om hvorvidt forskellige modeller skulle inddrages og hvordan rapporten skulle bygges op. Vibeke har vist sig at give stor værdi til teamet da hun guider os ved at give feedback og stiller kritiske spørgsmål, som vi reflekterer over og deraf bliver klogere.

Do:

Fastlagte mødetider

Inden projektstart havde vi ikke en decideret mødetid eftersom der er forskellige faktorer der spiller ind for hvert teammedlem, så vi aftalte bare at møde ind som man selv synes var fornuftigt. Vi har i løbet af pre sprintet lært at det har haft konsekvenser at alle ikke møder ind på samme tid, da vi ikke kan afholde vores daily scrum uden alle et fuldtalligt hold. Enkelte gange har vi afholdt daily scrum hvis vi manglede et teammedlem, dette har betydet at én fra teamet skal sætte tid af til at briefe den sent-ankomne, hvilket ikke har været optimalt. Vi har derfor valgt at sætte en tidsgrænse på at man møder ind på skolen inden kl 9.

Udeblivelse af medlem - fortsat gruppearbejde

Eftersom vi er studerende og nogle af os bor ude er det nødvendigt at supplere SU'en med et del-tidsarbejde. Vi har derfor i løbet af ugen, dage hvor vi ikke er fuldtallige, dette har resulteret i en dominoeffekt på de øvrige gruppemedlemmer som valgte at arbejde hjemme. Når vi arbejder hjemme er vi ikke agile og derfor bryder vi den udviklingsmetode vi har valgt at følge.

Vi har derfor valgt at droppe muligheden for at arbejde hjemme så længe der er én eller flere fremmødt på skolen.

Stop:

For lang ventetid mellem svar fra Belfast

Eftersom vores korrespondance med Business Mindset har vist sig været en udfordring med lange ventetider for hvert svar, har vi oftest stået tilbage i en situation hvor vi ikke følte vi kunne programmere noget da der har været problemer med at tilgå deres data.

For at mindske tidsspilde har vi valgt at lave fake data objekter for at kunne vise noget til vores eksamen.

Fra valideringsfasen i design sprintet fik vi en tilbagemelding fra Andrew og John som tydeligt illustrerer, hvorfor det som team er vigtigt at forstå kravene fra starten af og dermed at forstå krav ikke nødvendigvis sker på fem minutter.

Det viste sig at vores opfattelse omkring kravene og funktionaliteter til produktet var forskellige. På en eller anden måde havde vi fået to forskellige ideer om hvad der var prioritert højest ud fra den samtale vi havde haft.

John og Andrew ville gerne have at:

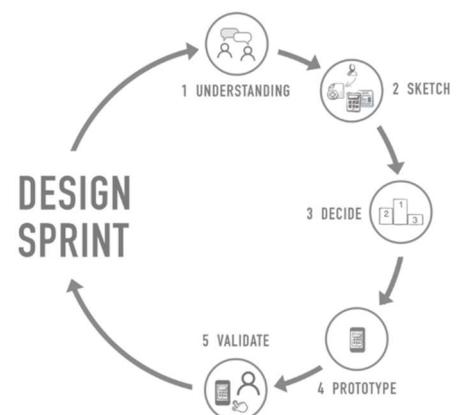
- Vores fokus til at starte med skulle ligge på hvordan et team blev oprettet og dernæst hvordan vi viser det i form af et dashboard.

Heldigvis fik vi denne feedback tidligt i processen, som gjorde at vi bare kunne smide det væk vi havde lavet på daværende tidspunkt og starte på et nyt design sprint for at forstå kravene endnu bedre end før.

Det også det som er essensen af et design sprint, det tager nemlig ikke lang tid at udføre, men udbyttet er stort og kan i sidste ende være med til at spare teamet for en masse udfordringer, fordi de netop ikke havde fået bekræftet om de har forstået kravene ordentligt.

Eftersom vi havde "misforstået" kravene i første omgang, arrangerede vi et møde om mandagen for at snakke om kravene til hvad et team er og hvad man vil bruge et team til.

- Et team består: Et team navn, X antal team member som skal kunne inviteres per e-mail. og grunden til der skal være et team, er så det kan vælges på dashboardet og vise indsigt i teamet.



UDVÆLGELSE AF USER STORIES

[Forfatter: Kasper, Ian, Jonas & Christoffer]

Før der kan udvælges user stories skal de estimeres og lægges i en prioriteret rækkefølge i product-backloggen, dernæst er det så simpelt at tage de vigtigste use cases fra toppen, men samtidig at tjekke om størrelse passer inden for den givende tidsramme (*tidsrammen kunne være et helt sprint, halvt sprint eller få dage*).

Estimering

Fremgangsmåden for estimering:

- Scrum masteren oprettede et planning poker projekt og uploadedede alle use cases hertil
- For at kunne estimere relativt til hinanden valgte vi et fælles udgangspunkt ved at fremhæve én bestemt use case, snakke om den og estimere den i fællesskab til en bestemt værdi.
- Scrum masteren læste use casene højt og alle medlemmer i teamet estimerede hver use case ift. fibonacci skalaen
- Da alle havde givet use casene en værdi, vendte scrum masteren point kortene og gjorde det visuelt tilgængeligt for alle, for at se hvem der havde givet hvad
 - Hvis der bred uenighed blandt nogle use cases, argumenterede de involverede for og imod. (Ofte opstod disse uenigheder i form af én eller flere medlemmer ikke havde forstået omfanget/sværhedsgraden af en use case)
 - Efter diskussionen var kommet til en ende, genstartede spillet om den samme use case og lod alle medlemmer komme med et revurderet svar.
- Da alle use cases var estimerede blev de prioriteret i rækkefølge efter systemkritisk funktionalitet (størst værdi for virksomheden) i vores product backlog (Hvis vi i teamet er i tvivl om hvorvidt hvad der giver størst værdi for virksomheden, skal det afklares med kunden).

Nu hvor vi har størstedelen af kravene skal de hver især estimeres så det gør os i stand til at planlægge sprints i forhold til hvor mange use cases der er realistiske at medtage. (*Nedenstående product backlog er i prioriteret rækkefølge*)

Story	Story Title	Score
1.	As a team leader I want to see a dashboard so that I can quickly get an overview of the composition in the team	13
2.	As a team leader I want to be able to create a team so that I can see team insight	10
3.	As a team leader I want to be able to differentiate various elements of the site through color coding so that it's easier to navigate	2
4.	As a team leader I want to read an in-depth report for the team, covering their test results so that I get an overview how well the team is structured	2
5.	As a team leader I want visual feedback so that it lets me interpret the test results more easily so that it's faster to use	3
6.	As a team leader I want to be able to add new members to an existing team so that relevant members can be assigned their relevant team	3
7.	As a team leader I want to be able to see a more insightful description of the individual	2
8.	As a team leader I want to be able to see a more insightful description of the individual members of the team, upon clicking on them so that I can have an overview of the team	4
9.	As a team leader I want to be able to delete team members so that non relevant members can be deleted	2
10.	As a team leader I want to be able to delete teams so that wrongfully made team can be deleted	2

Samlede mængde points i backloggen er: **43** (1 point svarer til 1 dag - *43 dage*). Hvis vi skal sætte det i perspektiv til hvad vi har til rådighed, så har vi 7 uger (*35 dage*) til deadline - De 35 dage inkludere alt fra at få teamet opsat, arbejde på realisering af use casene og til rapportskrivning.

Så som det ser ud på papiret, er det altså ikke muligt for os at nå i mål med alle kundes krav.

Det er kunden selvfølgelig blevet informeret om, da det kunne have haft spillet en rolle for hvordan der skulle/skal prioriteres ud igennem projektet.

KRAVÆNDRINGER D. 22/04-2019

[Forfatter: Ian, Jonas]

Hver mandag har teamets product-owneren som tidligere nævnt et “accept-brief” - på dette brief kom det frem at kunden prioriterede at der kunne oprettes et team som det første for at få noget funktionelt og fordi det er den byggesten der skal bygges videre på fremadrettet.

Kundens udtagelse resulterer i at vores produkt backlog måtte opdateres.

Story	Story Title	Score
1.	As a team leader I want to be able to create a team so that I can see team insight	10
2.	As a team leader I want to be able to differentiate various elements of the site through color coding so that it's easier to navigate	2
3.	As a team leader I want to see a dashboard so that I can quickly get an overview of the composition in the team	13
4.	As a team leader I want to read an in-depth report for the team, covering their test results so that I get an overview how well the team is structured	2
5.	As a team leader I want visual feedback so that it lets me interpret the test results more easily so that its faster to use	3
6.	As a team leader I want to be able to add new members to an existing team so that relevant members can be assigned their relevant team	3
7.	As a team leader I want to be able to see a more insightful description of the individual	2
8.	As a team leader I want to be able to see a more insightful description of the individual members of the team, upon clicking on them so that I can have an overview of the team	4
9.	As a team leader I want to be able to delete team members so that non relevant members can be deleted	2
10.	As a team leader I want to be able to delete teams so that wrongfully made team can be deleted	2

SPRINT PLANNING

[Forfatter: Christoffer]

På nuværende tidspunkt kender vi ikke teamets velocity og vi ved heller ikke hvor gode vores estimeringer har været, så for bedst muligt at planlægge vores sprint var der to ting vi gjorde:

- Den største faktor er at gå efter vores mavefornemmelse af estimeringerne, da vi har haft en eller anden underbevidst fornemmelse af hvor lang tid en bestemt use case måtte tage, på baggrund af rationel og logiske tanker samt tidligere erfaring med noget der kan relateres til use casen.
- Vi har 7 uger, hvor 1 uge er gået med pre sprint og en uge i vores sidste sprint (#3) vil blive afsat kune til at have fokus på at få færdiggjort rapporten, da rapporten i sig selv tæller super meget for vores eksamen - Det efterlader os med en ligning $25 \text{ points} / 5 \text{ uger} = 8,33 \text{ points pr sprint}$

Ved hjælp af ovenstående har vi derfor valgt at medtage use cases i sprint #1:

- As a team leader I want to be able to create a team so that I can see team insight - **10 points**

HVORDAN UDVIKLEDE VI

[Forfatter: Ian, Christoffer, Kasper & Jonas]

CSS Grid

Som en del af implementeringen af den første use case, skulle vi vælge hvordan vi ville implementere vores GUI, hertil valgte vi at bruge CSS grid layout. Vi valgte grid layout af flere grunde:

- Grid gør det muligt at lave et todimensionelt layout.
- Grid gør det nemt at gøre ens layout responsive og mobilvenligt hvis det skulle blive nødvendigt.
- Ved hjælp af grid areas kan vi holde vores design agilt og ekstremt nemt at ændre.
- Grid er et relativt nyt populært værktøj og nemt at implementere.

Vores design sprint har allerede bevist, at vores kundes krav og opfattelse af produktet kan ændre sig mange gange endnu, derfor er det vigtigt at vi nemt kan ændre layoutet af vores brugergrænseflade, uden de store besværigheder, dette giver grid os mulighed for. Med grid kan vi nemlig opdele vores layout i såkaldte grid areas, hvor vi altid kan flytte rundt på indholdet af disse areas. Se billedet nedenfor, her laver vi et grid layout i "team-creation-body":

```
.grid-container {  
    display: grid;  
    grid-template-columns: 1fr 1fr 1fr 1fr;  
    grid-template-rows: 1fr 1fr 1fr;  
    grid-template-areas:  
        "Left-Grid Top-Grid Top-Grid Right-Grid"  
        "Left-Grid Middle-Grid Middle-Grid Right-Grid"  
        "Left-Grid Middle-Grid Middle-Grid Right-Grid";  
}
```

Efter vi har defineret vores grid areas, kan vi bare tilkoble dem til vores divs på følgende måde:

```
.Middle-Grid {  
    grid-area: Middle-Grid;  
}  
.Left-Grid {  
    grid-area: Left-Grid;  
}  
.Top-Grid {  
    grid-area: Top-Grid;  
}  
.Right-Grid {  
    grid-area: Right-Grid;  
}
```

Hvis vi herefter vil flytte rundt på "invite-container", så kan vi bare gøre det ved at ændre "grid-template-areas".

HTML CSS

Grid er hvordan vores elementer på siden skal indordne sig og forholde sig til hinanden. Da de rammer var sat op, var næste opgave at lave alle vores HTML elementer og give dem en lille smule style:

Ethvert element på siden, har en funktion med henblik på det større billede af funktionaliteten. Som et udkast af vores design som set forneden, fungerede dette blot som en måde for os at kunne få visualiseret hvilke elementer vi har behov for.

Hvert element på siden har hver deres funktion, fra bare at vise plain tekst, tage tekst som input eller udføre noget funktionalitet ved et userklik.

Vi har tidligere erfaret at scenariet, hvor vi ikke planlægger men derimod bare springer ud i det og koder noget, som reelt (*i hvert fald i vores tilfælde*) resultere i at der bliver brugt tid på at

regne ud hvad næste skridt er og når vi støder på udfordringer (*hvilket vi helt sikkert vil*), er det svært at troubleshoot, for hvad var planen egentlig i første omgang?

Skridtet efter at have opsat at CSS og HTML var derfor at fastlægge hvad vores logik var og i hvilken rækkefølge ting skulle ske i.

Som tidligere nævnt i vores indledning, kender administratoren til medlemmernes faglige kvalifikationer på baggrund af den personlighedstest de har taget i forvejen.

Logikken:

- Team navnet skal vi bruge, så vi kan give all team members i teamet dens id. (se ER-diagram)
- Det skal være muligt efter indtastning af et team medlems email, at klikke og tilføje vedkommende til listviewet over teamets medlemmer, dette skal kun være muligt hvis mailen i forvejen eksisterer i Business Mindsets database. (*backend: til en intern list vi senere vil bruge til at give alle members det samme teamNameld*)
- Når knappen “Create Team” bliver klikket på, skal der ske flere ting
 1. Et team skal oprettes som det første (*kræver et team navn*) *POST kald*
 2. Den interne liste løbes igennem og ved hver memberID kaldes en andet *POST kald* (TeamId og MemberID)
 3. Til sidst vises der en alert der bekræfter at teamet er oprettet.

Normal User Flow:

Først vil admin (John) indtaste et team navn eller begynde at tilføje de team members der skulle være med i teamet (uanset hvad, vil bruger være nødsaget til at gøre det før der kan trykkes på “create team”) Create team klikkes og team med members bliver oprettet.

Database

Da det ikke har været muligt at få adgang til Business Mindsets database grundet en chance for at vi kunne komme til at slette værdifuldt data, eller på anden vis ødelægge struktur på deres database, er vi derfor nødsaget til at skabe vores egen mock database.

Det vil derfor ses at vores tabels ikke indeholder særlig meget data og grunden er at de kun indeholder det nødvendige data for at få vores feature til at virke.

Når kunden så skal tage vores feature i brug, handler det om at ændre connecting til deres database og tilføje de table vi har oprettet.

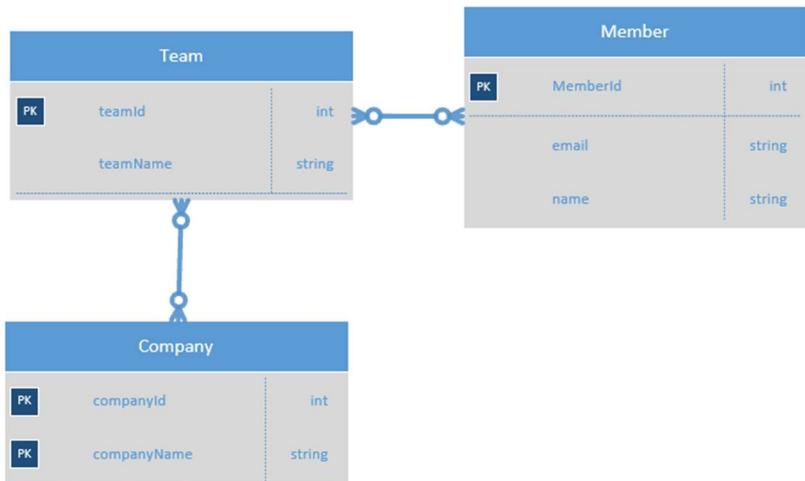
På nuværende tidspunkt ved vi hvad hjemmesides funktionalitet skal være, hvordan de skal udføres og hvilke informationer vi ønsker at få fat i.

Næste skridt var derfor at besluttede hvordan vi vil gemme alle de forskellige informationer og hvordan deres relation til hinanden skal være - ER diagram er et stærkt værktøj til dette.

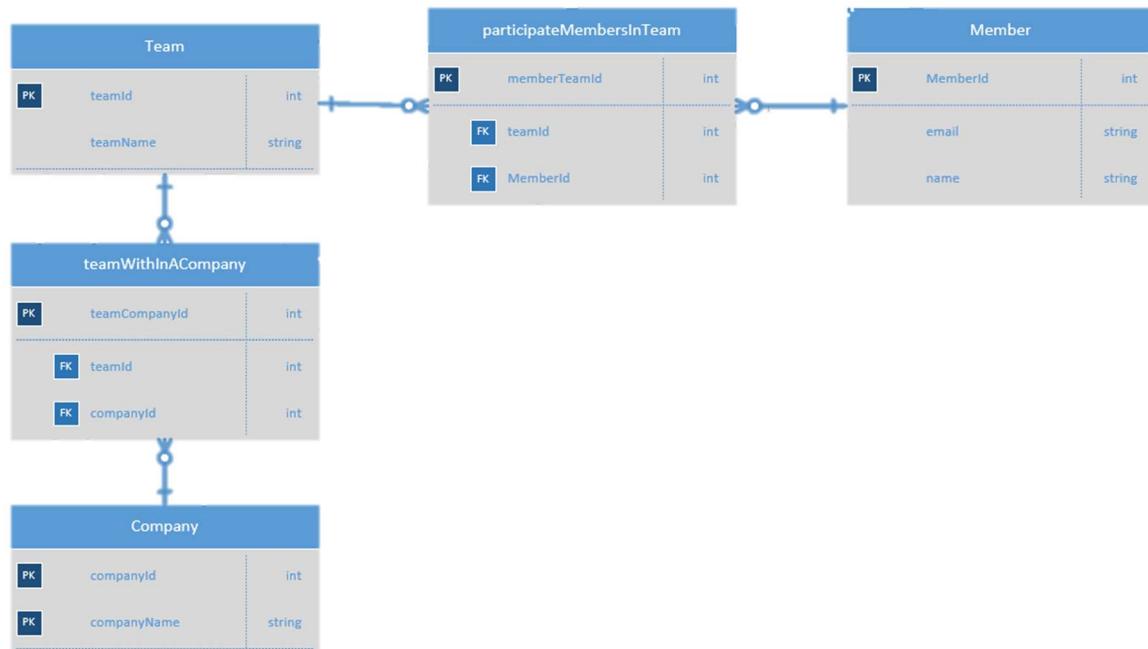
ER-diagram/DB-design

Vi har valgt at lave ER-diagram og DB design i et og samme diagram, for at spare tid og have det hele samlet - vi synes det giver god mening at man kan se relatioerne samtidig med de forskellige attributes hvert table har.

Som figuren til højre viser, har vi en multiplicitet, som er *mange til mange* – Det er fysisk umuligt at gemme data i tabellerne under sådan en relation, da det ville resultere i at en attribute ville have flere værdier pr række i databasen.



Løsningen: der oprettes en tabel imellem hvor de to tabeller hvor begge primærnøgle bliver foreign key idet nye table.



Senere vil vi også kigge på hvor god vores database er i forhold til duplikerede data og vedligeholdelse (se afsnittet om Normalisering).

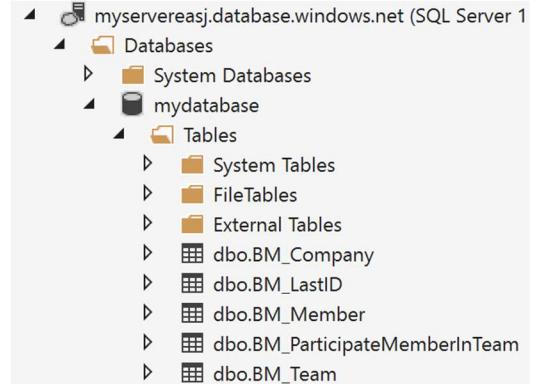
Azure

Der har ikke været behov for at oprette en helt ny database på Azure, da vi bare kan bruge den vi har fra vores tidligere projekter, vi oprettede derfor bare nye tables i denne.

I og med vi valgte at genbruge en tidligere database, var det vigtigt at navngive ordentligt, så man hurtigt og nemt kunne finde sine tables, derfor navngav vi dem "BM_tablenavn".

(BM, står selvfølgelig for **Business Mindset**)

At have ovenstående diagram lige ved hånden, gjorde det super nemt at oprette vores tables i Databasen, vi viste:



- Hvilke attributes der skulle være primærnøgle og foreign keys
- Hvilke attributes der skulle hvor i hvert table
- Hver attributes type

Name	Data Type	Allow Nulls
comapnyId	int	<input type="checkbox"/>
companyName	nchar(50)	<input checked="" type="checkbox"/>

Funktionalitet

Et view og en database er ikke meget uden også noget tilsvarende funktionalitet der kan gøre brug af dem, derfor var det selvfølgelig næste skridt.

Vi startede med noget helt simpelt funktionalitet, hvor vi griber fat i HMTL elementerne: Listview, textfield og add-knappen med "*document.getElementById*" hvorefter vi så kan smide den indtastede værdi over i listview ved tryk på knappen.

```
... ...
@@ -2,11 +2,17 @@ var url = "www.google.com";
2   2   var request = new XMLHttpRequest()
3   3   request.open('GET', url, true)
4   4
5 + /* Binding to view fields */
6   6   const insert_team_name = document.getElementById('name')
7   7   const find_team_members = document.getElementById('email_textbox')
8   8   const add_member_button = document.getElementById('add_member_button')
9 - const member_list = document.getElementsByClassName('member_list')
10  9 + const member_list = document.getElementById('member_list')
11 10
12 - function addItem(){
13 -
14 11 + /* Add new member function */
15 + function addMember(){
16 12 +     var ul = document.getElementById("member_list");
17 13 +     var li = document.createElement("li");
18 14 +     li.setAttribute('id',find_team_members.value);
19 15 +     li.appendChild(document.createTextNode(find_team_members.value));
20 16 +     li.appendChild(document.createTextNode(find_team_members.value));
21 17 +     ul.appendChild(li);
22 18 }
\ No newline at end of file
```

Skridtet videre, var at gå i gang med at kode vores REST (som tidligere beskrevet oppe i afsnittet om HTML og CSS i "viewets logic")

- POST - Opret et team (som bare var et referencenavn til at gruppere members og team sammen)
- POST - memberToTeam (gruppere members og et team sammen)
- GET - memberInTeam (henter alle members i et specifikt team)
- GET - getTeam (henter alle oprettede teams)

- GET - getTeamByName (henter et specifikt team pga. det navn)

Interface - Her har vi erklæret vores metode signatur og hvilke type metode kaldet er (*GET*).

```
/// <summary>
/// Allow admin to get a list of all created teams.
/// </summary>
/// <param name="team"></param>
[OperationContract]
[WebInvoke(Method = "Get", ResponseFormat = WebMessageFormat.Json, UriTemplate = "team/")]
List<Team> GetAll();
```

Service - Her implementere vi vores interface og giver vores metoder en krop.

Der er skrevet en query der kan hente team id og navn fra det table der indeholder disse værdier (nemlig BM_Team). Så bliver der åbnet en forbindelse til vores database på Azure, vha. en connectionstring (*den kan findes inde på Azure*).

Der laves et SQLcommand objekt, så vi kan kalde vores query i vores database, hver række af data i databaser bliver læst og gemt i en liste, indtil der ikke er flere rækker i databasen.

```
public List<Team> GetAll()
{
    //listen der skal vises i browseren
    List<Team> OList = new List<Team>();

    // dette er en connection string som valger hvilke tabels vores data skal ind i.
    const string sqlstring = 'SELECT teamId, teamName FROM dbo.BM_Team';

    using (var DBconnection = new SqlConnection(ConnectionString))
    {
        DBconnection.Open();
        var sqlCommand = new SqlCommand(sqlstring, DBconnection);

        // Her benytter vi vores sqlconnection og sqlCommand til, at aflese data fra databasen.
        using (SqlDataReader reader = sqlCommand.ExecuteReader())
        {
            while (reader.Read())
            {
                Team team = new Team();
                team.teamId = reader.GetInt32(0);
                team.teamName = reader.GetString(1).Trim();
                //trim removes whitespaces.

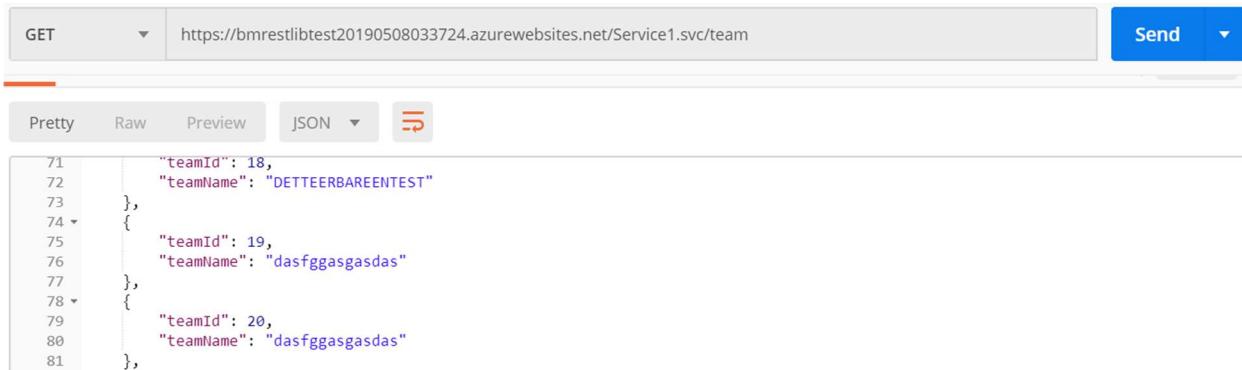
                OList.Add(team);
            }
        }
        return OList;
    }
}
```

Testing

For at sikre os produktet vi leverer til kunden, er af højest kvalitet og i det hele taget fungerer hensigtsmæssigt, har vi valgt at teste følgende områder:

Test af REST

Når vi har arbejdet med at skulle sikre os at vores REST og dens metoder virker hensigtsmæssigt har vi benyttet os af Postman, da den kan kalde metoder fra REST via URL'en hosted på vores Azure server.



```
71     "teamId": 18,
72     "teamName": "DETTEBAREENTEST"
73   },
74   {
75     "teamId": 19,
76     "teamName": "dasfggasgasdas"
77   },
78   {
79     "teamId": 20,
80     "teamName": "dasfggasgasdas"
81 },
```

Ovenstående billede viser et GET kald på vores REST service. Ved at teste vores GET metode kræver det at vores POST metode også har fungeret, og som aflæst ud fra billedet kan vi konkludere med denne simple test, at vores funktion virker i og med at vores dummy data teams kommer frem med forskellige Id'er

Test af hjemmeside

Når vi har foretaget os af test af hjemmesiden, startede vi med at benytte os af manuel testing i form af brugertest. Her undersøgte vi, som en potentiel bruger ville gøre, om alle knapperne på hjemmesiden virkede som de skulle.

Eftersom vi indså at processen var meget gentagelig når vi ændrede i koden, besluttede vi os få at benytte os af Selenium.

```
[Test]
public void CheckIfMemberIsAdded()
{
    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(5000));
    IWebElement emailDummy = driver.FindElement(By.Id("email_textbox"));
    emailDummy.SendKeys("ianfrost0@gmail.com");

    driver.FindElement(By.Id("add_member_button")).Click();

    wait.Until(ExpectedConditions.ElementExists(By.Id("ianmfrost0@gmail.com")));

    IWebElement MemberList = driver.FindElement(By.Id("member_list"));

    bool NewElementExist = MemberList.FindElement(By.Id("ianmfrost0@gmail.com")).Displayed;

    Assert.AreEqual(true, NewElementExist);
}
```

Ovenstående kode viser en test case der tjekker om et nyoprettet medlem bliver vist i listviewet. Eftersom vores funktionalitet på nuværende tidspunkt er meget minimal, er det også minimalt hvad der kan automatiseres.

Ved at benytte os af Selenium kommer vi på sigt til at spare tid da vi kommer til at skulle teste flere funktioner end på nuværende tidspunkt. Alle test cases bliver markeret enten ved rød farve (hvis

testen er fejlet) eller grøn farve (hvis testen fungerer som forventet), dette giver også teamet et bedre overblik, når vi på senere tidspunkt har flere funktioner der skal testes, viser test case navne hvilke funktioner der har ændret sig, hvis der har været ændring i koden.

▲	BM_Selenium (1)	9 sec
▲	BM_Selenium.Testcases (1)	9 sec
▲	CreateTeamPage (1)	9 sec
	CheckIfMemberIsAdded	9 sec

Test af database

For at tjekke at vores table er korrekt opsat, giver vi dem alle sammen noget test data, der svarer til de værdier de vil komme til at indeholde når det bliver taget i brug.

	memberId	memberEmail	memberName	
►	1	lanmfrost0@gmail.com	Ian Frost	...
	2	KasperKofoed@live.dk	Kasper kofoed	...
	3	Chris31@live.dk	Christoffer Dunk	...
	4	Jonasmanley@hotmail.com	Jonas Pedersne	...
*	NULL	NULL	NULL	

På den måde kan vi tjekke at hver' attribute kan holde den rigtige værdi og at vores ID bliver auto inkrementeret.

Derudover kommer dataen også til at spille en rolle senere, når vi for alvor skal teste vores færdige funktioner.

Normalisering af data

Vi har kigget på normalisering af vores database for at gøre den så effektiv som mulig når der tilføjes, opdateres og slettes data, men også for at minimere redundant data, dette betyder at vi ikke vil have samme oplysninger gemt flere forskellige steder i databasen.

Problemerne med redundant data er:

- Performance - bliver nedsat når man fjerner, opdaterer eller tilføjer data, da data vil være flere steder og der ikke er nogen rød tråd.
- Vedligeholdelse - Det bliver nemmere at vedligeholde når man kun skal ændre ét sted i databasen fremfor flere steder. (*Indenfor programmeringsverdenen følger man samme princip, DRY, som gør det nemmere at vedligeholde vores kode.*)

Normal former

Normalisering organiserer data i databasen ved at oprette tabeller og relationer mellem disse tabeller på baggrund af følgende regler:

1NF (First normal form):

For at vores database skal opfylde 1NF skal følgende betingelser være opfyldt:

- Tabellen skal indeholde single værdier:
- Entries skal være i kolonne af samme type
- Rækkerne identificeres med en primary key (PK)

Nedenstående tabel indeholder 2 kolonner (EmailId, Name).

Vores kolonne "Name" indeholder samme entries værdier og ikke mere end en - Vi kan ikke have to personer stående i et felt.

Der er indført en primær nøgle, som har til formål at skelne mellem navne som ligner hinanden, som ses på eksemplet, derfor er det vigtigt at have en primary key (Email Id)

Email ID (Primary key)	Name
ian@hotmail.com	Ian Frost
ianJens@hotmail.com	Ian Frost

2NF (Second normal form):

Hvis vores tabels skal opfylde 2NF skal følgende krav opfyldes:

- Alle krav fra 1NF skal være opfyldt
- Hvis tabellen er sammensat nøgle, skal alle de andre felter være afhængig af nøglen

Nedenstående tabel er en sammensat tabel mellem Team & Company, der hver har givet deres primary nøgle til tabellen som foreign key. Ved at disse foreign keys er afhængige af primary key gør den 2NF.

CompanyTeamId (primary key)	TeamId (foreign key)	CompanyId (foreign key)
2	2	2
1	7	7

3NF(Third normal form):

Hvis vores database tables skulle opfylde 3NF skulle følgende betingelser være opfyldt:

- 2NF kravene skulle alle være opfyldt
- Der må ikke være felter som er indbyrdes afhængige - i tilfælde af dette brydes tabellen yderligere ned.

Eftersom vores tabeller ikke indeholder yderligere indbyrdes afhængigheder af værdier og alle vores felter er afhængige af en primary key, opfylder vi IKKE kravene for 3NF.

Et klassisk eksempel på 3NF vil være at man havde et table hvor (**id, navn, email, postnummer og bynavn**) og her vil postnummer og bynavn være indbyrdes afhængige af hinanden og derfor kan man bryde dette table op i to: (**postnummer / bynavn**) og (**navn / email**).

Se nedenstående tabel for visuelt eksempel:

Brugere (2NF)

Id	Navn	Email	PostNr	Bynavn
1	Jonas	Jonas@hotmail.com	4000	Roskilde
2	Ian	Ian@hotmail.com	4200	Slagelse
3	Christoffer	Dunk@hotmail.com	4261	Dalmose

Brugere (3NF)

Id	Navn	Email
1	Jonas	Jonas@hotmail.com
2	Ian	Ian@hotmail.com
3	Christoffer	Dunk@hotmail.com

PostNr	Bynavn
4000	Roskilde
4200	Slagelse
4261	Dalmose

Som tidligere nævnt i rapporten er vores endelige produkt en feature til en allerede eksisterende, kørende og funktionel applikation som har alt nødvendigt data for at fungere. Vi skal ikke replikere dette, men bruge det data som er nødvendigt for at vores feature fungere optimalt.

VERSIONSSTYRING

[Forfatter: Christoffer, Jonas]

Oprettelse af ny branch

Da vi har valgt at bruge git som versionskontrol program, skal vi selvfølgelig også opsætte vores working branches. Vi har gjort som man vil gøre i det i den professionelle verden, hvor vi har en master branch der indeholder alt godkendt og færdigt udviklet kode, hvor vi ved siden af, har vores working branches som vi udvikler på. Når en working branch så er færdiggjort og godkendt bliver den merged med master.

Så efter vi havde oprettet projektet (en tom mappe med navn), skulle vi oprette vores første working branch som hedder "team_page", dette gør man blot ved at skrive følgende i vores git bash:

```
Dunk@DESKTOP-FSH6TDS MINGW64 ~/Desktop/eksamensprojekt (master)
$ git branch team_page|
```

Efter dette har man kun oprettet en ny branch lokalt, derfor pusher man også den nye branch til gitlab så resten af gruppemedlemmerne også kan hente den nye branch. Dette gør man således:

```
Dunk@DESKTOP-FSH6TDS MINGW64 ~/Desktop/eksamensprojekt (master)
$ git push -u origin team_page|
```

Herefter kan alle i teamet se branchen på GitLab hvor vi har valgt at opbevare vores projekt.

The screenshot shows a list of active branches. The 'master' branch is listed first, marked as 'default' and 'protected'. It has a merge commit from the 'team_page' branch. The 'team_page' branch is marked as 'merged' and has a recent commit. There are buttons for 'Merge request', 'Compare', and a trash can icon.

På denne måde lykkes det os at holde vores endelige og godkendt version fra vores udviklende version, som muligvis kan støde ind i problemer. Hvilket gør at vi minimere risikoen for tab fungerende kode.

Push og pull

Ligeledes som når vi pusher en ny branch, skal vi også pushe hver gang der implementeres en opgave. Dette gennemføres gennem tre forskellige commands:

1. git add -A

I det første step tilføjer vi alle de redigeringer vi vil have med i vores commit, ved at skrive -A tilføjer man alle ændringer og ikke bare en specifik ændring.

2. git commit -m "message der beskriver dette commit"

I step to laver man sit commit og giver det en passende besked med, så den let kan identificeres.

3. git push

Her i det sidste step pusher vi endelig vores nye commit, så vores remote branch også er opdateret med de ændringer vi har lavet.

Efter at ens commit er pushet kan de andre gruppemedlemmer opdatere deres lokale version med remote versionen ved hjælp af denne command:

git pull

Når vi både kan push og pull kan vi gøre det ønskede nemlig, at vi let kan arbejde på det samme projekt hele gruppen og vi kan nemt review vores nye kode og godkende det.

Merge

Den sidste proces som nye implementeringer løber igennem inden de er synkroniseret med master version, hedder merge. Det er i denne proces vi kontrollere den nye kode og vurdere om vi er tilfredse med den, som gør på GitLab.com hvor vi har oprettet vores projekt. GitLab har en unik brugergrænseflade, hvor vi let og overskueligt kan kontrollere vores commits, eksempelvis kan vi let browse igennem alle commits, som ses nedenfor:

Her kan vi nemt scroll gennem alle vores ændringer, så hvis vi pludselig mangler noget eller det ikke virker, så kan vi nemt spore fejlen og genskabe den tidligere version.

Udover dette kan vi se alle ændringerne som branchen indeholder tilsammen:

Discussion	0	Commits	24	Changes	6
20 May, 2019 2 commits					
	alert when memberToTeam is created	Jonas Pedersen authored 3 days ago	2100ca2b		
	create team function completed	Christoffer authored 3 days ago	ec31fc6e		
16 May, 2019 1 commit					
	small adjustments	Christoffer authored 1 week ago	afc0cc74		
15 May, 2019 3 commits					
	condition added to addmember	Christoffer authored 1 week ago	b65e997d		
	add member button is verifying the id with the database	Christoffer authored 1 week ago	908c261b		
	API call Get	Christoffer authored 1 week ago	d6fbbed9a		

The screenshot shows a GitHub pull request interface. At the top, there are tabs for 'Discussion 0', 'Commits 24', and 'Changes 6'. Below this, a search bar says 'Changes between latest version and master'. To the right, it shows '6 Files' with 494 additions and 2 deletions, along with a gear icon and dropdown menu. A 'Filter files' input field is present. On the left, a sidebar lists files with their change counts: .vscode (0), settings.json (+3 -0), grid.css (+90 -0), index.html (+49 -2), login.html (+14 -0), scripts.js (+118 -0), and style.css (+220 -0). The main area displays the 'grid.css' file content:

```

1 + .header {
2 +   display: grid;
3 +   grid-template-columns: 1fr 1fr 1fr;
4 +   grid-template-areas:
5 +     "title title title"
6 +     "create-team edit-team view-team";
7 + }
8 +
9 + .title {
10 +   grid-area: title;
11 +   height: 20vh;
12 + }
13 +
14 + .border-navigation-buttons {
15 +   grid-column: 1 / 4;
16 +   grid-row: 2;
17 + }

```

Det er her vi kan review koden og vurdere om det er klar til at blive merged med master branchen. Hvis alt ser tilfredsstillende ud bliver der lavet en merge request, som derefter bliver godkendt. Derimod hvis det ikke er tilfredsstillende og det vurderes at implementeringen skal laves om, så laver vi bare nye commits indtil vi er tilfredse med resultatet.

KRAVÆNDRINGER D. 29/04-2019:

[Forfatter: Jonas & Ian]

Efter mandagens “accept-brief” kom det frem at kunden prioriterede login og muligheden for at oprette en virksomhed.

På baggrund af denne samtale kom der altså to nye use cases ud:

- As a admin i want to be able to login so I can see all the different pages and can create a team and company.
- As a admin i want to be able to create a team which is just a name so i can select it from a drop-down i the team creations site.

Nu hvor der er kommet nye user stories ind, er det nødvendigt at spille *PlaningPoker* om disse og efterfølgende smide dem i product backloggen afhængig af deres prioritet (Kunden sagde at disse krav var højt prioritet)

De nye stories vil derfor være dem vi vil tage med i sprint #2, om vi tager begge med er afhængig af teamets velocity

Story	Story Title	Score
1.	As a team leader I want to see a dashboard so that I can quickly get an overview of the composition in the team	13
2.	As a team leader I want to be able to create a team so that i can see team insight	10
3.	As a team leader I want to be able to differentiate various elements of the site through color coding so that it's easier to navigate	2
4.	As a team leader I want to read an in-depth report for the team, covering their test results so that I get an overview how well the team is structured	2
5.	As a team leader I want visual feedback so that it lets me interpret the test results more easily so that its faster to use	3
6.	As a team leader I want to be able to add new members to an existing team so that relevant members can be assigned their relevant team	3
7.	As a team leader I want to be able to see a more insightful description of the individual	2
8.	As a team leader I want to be able to see a more insightful description of the individual members of the team, upon clicking on them so that I can have an overview of the team	4
9.	As a team leader I want to be able to delete team members so that non relevant members can be deleted	2
10.	As a team leader I want to be able to delete teams so that wrongfully made team can be deleted	2

OPSUMMERING AF SPRINT #1

[Forfatter: Kasper, Ian, Christoffer & Jonas]

Hvad har vi lavet?

Efter vores første sprint er overstået kan vi lige opsummere hvad produktet er i stand til på nuværende tidspunkt.

Som tidligere nævnt i rapporten har vi lavet alle de små ting der skal til bag facaden for at få hjemmesiden til at fungere.

Vi har:

- Opsat CSS grid, designet vores HTML og CSS,
- Oprettet vores database med tilsvarende tabels i forhold til de værdier der skal gemmes.
- Gjort så vores knapper på vores frontend side har kendskab til vores listview, så vi kan tilknytte dem nogle funktionalitet.
- Startet på at kode vores REST web service, så vi var i stand til at kalde metoderne fra vores hjemmeside. (GET & POST var især i fokus).
 - Den eneste metode der er blevet lavet er getTeams - som henter all created teams som er i vores database.

Hvorfor har vi ikke nået alt det vi planlagde i sprint #1?

Vores første sprint bød på en del flere forhindringer end vi kunne havde forudset. Vores kontakt med Business Mindset var i det hele taget lidt vanskelig til at starte med, da vores e-mails endte i deres spam-mappe, der gik mange timer før vi fik svar på spørgsmål og det betød at vi var meget længe om at få adgang til deres materiale (*website, filer og API*), så vi kunne se hvor meget data vi kunne hente fra deres system.

Dummy-data

Fordi det tog så lang tid, havde vi i mellemtiden besluttet os for at lave *dummy-data*, så vi kunne komme videre og ikke bare spilde vores tid med at vente på feedback - og som i sidste ende gjorde vi kunne udvikle noget funktionelt som kunden kunne evaluere.

Ydermere bød sprintet på meget research, eftersom vores use case krævede en del javascript og ny syntax som vi ikke havde stiftet så meget bekendtskab med før, selvom research selvfølgelig var inkluderet i vores estimeringer fra vores planning poker - så var der stadig meget mere end hvad vi lige havde regnet med i første omgang.

Kontakt med business mindset evaluering

Eftersom at vores kunder både John såvel som Andrew, begge har meget travle hverdage med en bunke forskellige andre projekter - der på nuværende stadig har en væsentligt højere prioritet end dette. Er det ikke en overraskelse at de sætter mindre tid til side til dette end de andre projekter da deres fokus selvfølgelig skal være på reelle kunder.

Da vi stadig er meget unge indenfor lige netop denne branche kan der heller ikke lægges de helt samme forventninger til arbejdstid og kvaliteten indenfor en given tidsramme i forhold til hvis de havde betalt professionelle til at udføre arbejdet for dem. Det er derfor indforstået at vi arbejder på projektet for dem gratis med henblik på at kunne hjælpe dem med at hjælpe os til at lære mere omkring branchen. Da det netop bliver udviklet gratis for dem, sikrer dette også at der ikke er nogen ressource relaterede risici for dem. Som et sideprojekt, har vi desværre også lidt følt konsekvenserne af at vi er lavere prioriterede i forhold til hvor ofte vi har kunne have en korrespondance med dem. Gennem forløbet har vi haft perioder hvor manglen på feedback har resulteret i at vi i bund og grund, endte med at arbejde lidt i blinde, i håb om at ramme hvad det er de reelt set har af forventninger efter et givent sprint. Dog kan dette også vendes til en positiv ting, da det er en reel problemstilling på det rigtige arbejdsmarked. Hvad stiller vi lige op når kunden enten ikke er gode nok til at være konkrete med deres krav, slet ikke har krav eller er dårlige til at give feedback på udført arbejde? Vi har måtte indordne os efter en forventning om ikke at få respons med det samme, når vi kontakter dem. Med en forventning på hvert fald en dag eller to mellem deres svar og med møder der bliver udskudt, må vi kunne aktivere os selv for ikke at ende med at sidde fast og vente på feedback. Dette har vi blandt andet haft gjort som beskrevet i punktet ovenfor i "**Dummy-data**".

Den ideelle process er selvfølgelig en process der er prioriteret, i vores øjne er det essentielt for os at kunne få regelmæssig feedback. Både som en måde for os at kunne holde bedre overblik over hvad det er der forventes af os. Men også med henblik på at kunne motivere teamet bedre, med langt mere regelmæssige deadlines i form af daglige møder, eller møder hver anden dag. Har vi en egentlig ting at tage og føle på, for ligesom næsten at kunne skære ud i pap hvad det helt præcist er vi har at skulle lave på daglig basis - dette er med henblik på en process der er adapteret til et team med manglende eller begrænset erfaring.

Hvad er planen fremadrettet?

Eftersom vores use case var meget omfattende nåede vi desværre ikke i mål med den i vores første sprint.

Men eftersom vi er meget tæt på at være i mål med use cases, måtte bare medtage den i sprint #2, efter at vi har revurderet den i planning poker, da den selvfølgelig er væsentlig mindre end i starten.

Vores REST web service mangler ikke meget for at metoderne virker og derefter skal den kobles til vores javascript.

SPRINT REVIEW:

[Forfatter: Ian & Christoffer]

Fredag d. 10/05-2019 afholdte vi vores sprint review med John og Andrew. Vi fremviste hvad vi havde lavet i første sprint og her var det vigtigt for teamet at vi fik en bekræftelse på at vi var på rette vej. Selvom det var begrænset hvad vi havde at vise frem, rent software mæssigt, var det en topprioritet af få feedback på hvad vi havde så vi havde en chance for at imødekomme så mange ændringer som muligt.

John og Andrew var begge enige om det simple design var en god idé og vores funktion virkede som den skulle i forhold til kravene. De ønskede heraf en user journey fra oprettelse af teams & company samt de forskellige sider der forekommer i vores applikation, indenfor de næste dage. Så de kunne se hvad vi havde tænkt os i grove træk.

Ydermere, er login nu i deres øjne også blevet en højt prioriteret opgave i næste sprint, da de ikke ønsker at alle skal kunne tilgå vores data via deres URL.

Til sidst i samtalen havde vi en spørgerunde for begge parter, vi ønskede at få adgang til deres API, da vi ikke kunne finde den specifikke fil i deres system, så vi kunne trække det nødvendige data.

Deres spørgsmål til os var hvad næste skridt i processen var, samt om vi havde forstået logikken bag inddle points til diverse teams. Vores svar hertil var at vores første user story var langt mere omfattende end forventet, og derfor var det nødvendigt at prioritere at færdiggøre den i næste sprint.

- Spørgsmål til os
 - Fremtidige plan (team creation done)
 - Forstår vi logikken bag point inddeler teams? (2 excel sheets vi har fået udleveret)
- Manglende informationer blev indsamlet
 - Vi skal have adgang til API
 - Kan vi trække alt den nødvendige data (ja)

Sprint #2

SPRINT RETROSPECTIVE

[Forfatter: Ian, Jonas, Kasper & Christoffer]

Som i starten af vores Sprint #1, hvor vi lavede et retrospective over vores pre sprint. Laver her et retrospective til sprint #1.

Keep:

Daily scrum

Giver fortsat værdi for holdet og derfor ønsker vi at fortsætte med at benytte os af det, da det skaber overblik over hvad folk har af arbejdsopgaver.

Sit together

Tidligere i forløbet havde vi en tendens til at arbejde hjemme hvis der var nogen fra gruppen der udeblev grundet sygdom eller arbejde. Efter vi fik vendt dette i vores pre sprints retrospective valgte vi at indføre **sit-together** altså arbejder vi sammen og møder op selv om andre skulle være syge eller på arbejde, dog indforstået at der selvfølgelig stadig kan være andre ting der forhindrer fremmøde. Det har vi valgt at indføre i næste sprint, da det har gavnnet med sparringer blandt teamet i forhold til arbejdsopgaver - og det fungere meget bedre, det viser sig at teamet er teamet mere effektivt på skolen også selvom der måtte være et medlem der ikke er der.

Dummy data

Vi indså hurtigt i pre-sprint at vores korrespondance med Business Mindset gjorde det problematisk at komme i gang med arbejdsopgaverne, da der var flere dage mellem svarene. Derfor tog vi beslutningen om at benytte os af **Dummy data** så vi kunne skabe noget fungerende software som vi kunne vise kunden og derfor udnytte tiden.

Informative workspace

Vi har valgt at fortsætte med at benytte os af **informativ workspace**, da det er en super god måde at få visualiseret vigtige ting for hele teamet, så alle har en forståelse for hvad hvert medlem arbejder på.

Vi har haft stort udbytte af det ved vores daily scrum mødet, fordi det giver mødet struktur.

Stop:

Ud fra vores pre sprint retrospective blev vi enige om at vi ville **fastlægge mødetiderne**, så alle medlemmer mødte inden kl 9. Ud fra forskellige faktorer hos diverse teammedlemmer har vi ikke opfyldt denne ændring, derfor ser vi ikke nogen nytte i at fastlægge mødetider, men blot fokusere på at være produktive når vi er fuldtallige.

Do:

Kanban board

Vi har i løbet af projektet erfaret af vores user stories kan være omfattende og derfor kan de virke uoverskueligt i forhold til hvilke tasks der er påbegyndt, færdige eller mangler. Ved benyttelse af **Kanban boardet** inddeler vi alle tasks under hver user story. Kanban boardet indgår som en del af vores informative workspace.

Dagens motto

Vi har måske opdateret dagens motto 2 gange om ugen, hvilket ikke lever op til hvad vi planlagde (*nemlig hver dag i forbindelse med daily scrum*) - *dog har det stadig været med til at give noget liv til vores rum og fjerne den meget statiske og kedelige atmosfære der er i et hvidt og indelukket mødelokale med opgaver hængende på væggen.*

2 x 20 min. homework

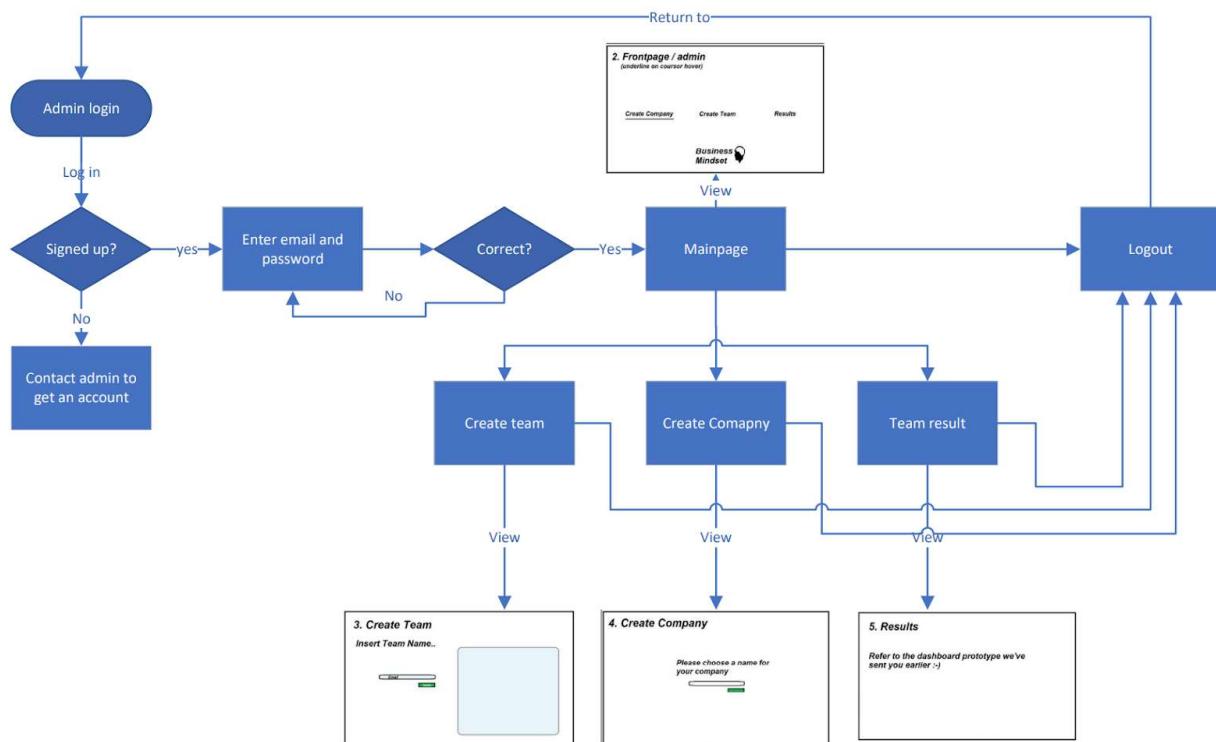
Hvis målet for rapportens indhold skal nås inden vi begynder at sætte os for meget under press og overarbejde, har vi valgt at indføre at man som minimum skal kigge på rapporten i tyve minutter derhjemme to gange ugentligt. Dette er med til at sikre vores rapport bliver tjecket igennem flere gange og kan i sidste ende finde fejl man har overset tidligere.

USER JOURNEYS

[Forfatter: Jonas]

Som efterspurgt af John og Andrew i vores sprint review i sprint #1, udarbejdede vi en user journey, som kunne illustrere vores tanker omkring hvordan systemet skulle opbygges herunder hvilke sider der skulle have kendskab til hinanden.

Dette har til formål at Andrew og John kunne komme med en evt. bekræftelse af vores tanker.



SPRINT PLANNING

[Forfatter: Ian & Kasper]

I forlængelse med vores sprint review fandt vi ud af vores første use case var meget omfattende, derfor ser vi os nødsaget til at inddrage den i næste sprint. Eftersom de fleste af vores tasks var klaret, har vi revurderet den i planning poker så den har en mere passelig værdi til vores sprint. Ydermere, nævnte John at loginfeaturen var prioriteret og derfor indgår denne use case også i dette sprint.

Nedenstående billede viser vores trello board for sprint 2, her er følgende use cases med:

- As a team leader I want to be able to create team so that I can see team insight - 3 point
- As an admin i want to be able to login so I can see different pages and create team and company - 7 points

Man kan undre sig lidt over hvorfor vi tager 10 points med i sprint #2, da vi ikke kunne klare denne opgave i sprint #1 hvor vi også havde 10 points med.

Grunden er at vi under vores re-estimering, var ret splittet på hvor lang tid der var tilbage af den use case som vi fik færdiggjort: to stemte 1p og 2 stemte 3p

Konklusion var at vi satte barren højt og viste det sig så, at gå hurtigere ville det selvfølgelig være foretrukket og det klart bedste scenarie.

The image shows a digital Kanban board interface. At the top, there are tabs for 'Programmering', 'CodeSharks Free', 'Team Visible', and 'Invite'. Below these are five columns: 'Backlog', 'Sprint 1 - 29-04-2019 / 10-05-2019', 'In progress', 'Ready for review', and 'Sprint 1 Done'. Each column has a 'Add a card' button. The 'In progress' column contains two cards:

- Sprint 1 - 29-04-2019 / 10-05-2019**
As a team leader I want to be able to create a team so that I can see team insight - 15 points (sprint #2 = 3) - // Torsdag.
11/13
- In progress**
As a team leader I want to be able to differentiate various elements of the site through color coding so that it's easier to navigate - 2 points
As a team leader I want to be able to see the team members personality tendencies so that I can better guide them into their roles - 4 points
0/7

Below the 'In progress' column, there is a '+ Add another card' button.

HVORDAN UDVIKLEDE VI

GUI DESIGN

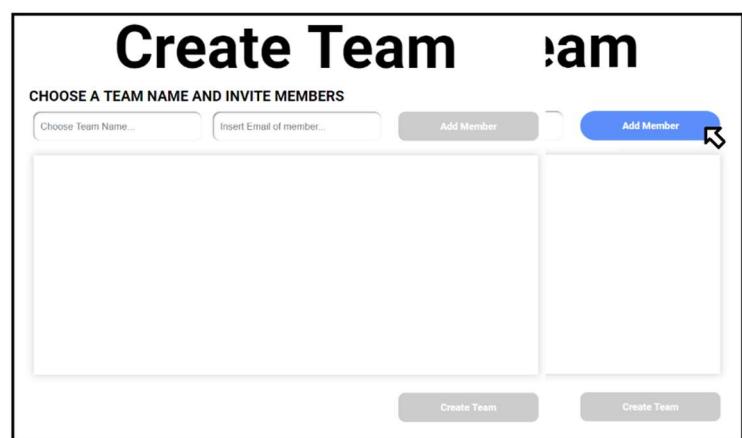
[Forfatter: Kasper]

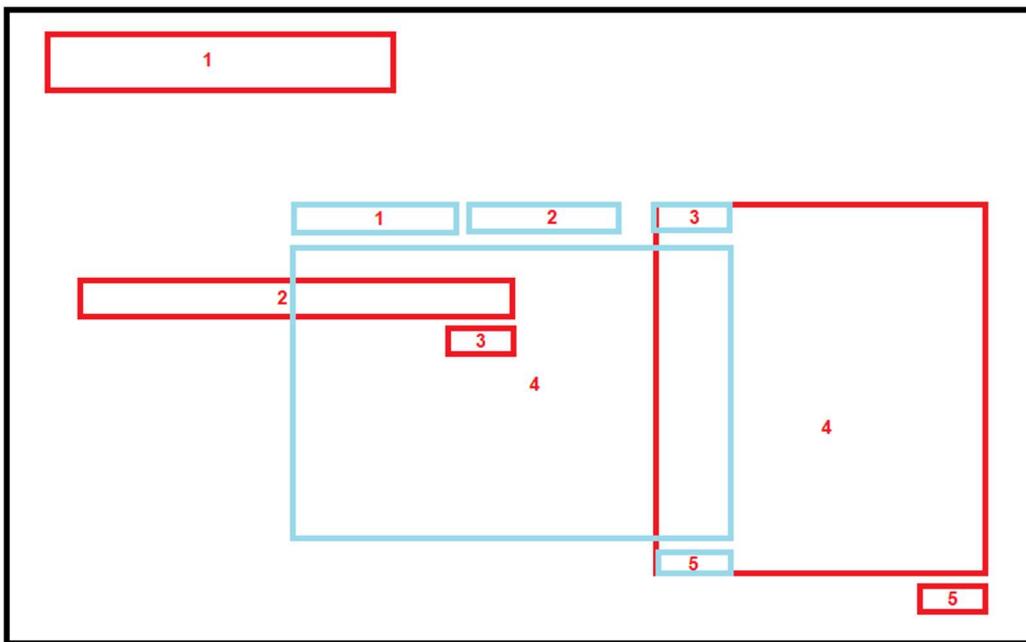
Grundet alderen på vores produkts primære brugere, har vi set det som værende en god ting at holde vores UI-design simpelt og intuitivt. Med feedback fra Business Mindset, fandt vi frem til et prototypedesign som vi kunne bruge som springbræt hen til en mere endelig version af vores UI. Set nedenfor er dette vores tidligere version af blandt andet, vores **Create Team** page.



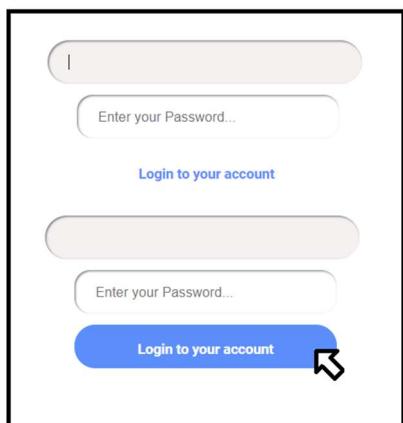
Vi valgte dog at ændre på dette, da vi ikke følte at det nødvendigvis var hvad vi ønskede med vores design. Det første design er alt for fragmenteret og opdelt rundt omkring på siden. Øverst indtaster vi team navn, dernæst invitation af medlemmer og så kigger vi over til højre for en oversigt. Frem for dette, valgte vi at samle elementerne mod midten af siden da vi føler at det giver bedre mening ikke at skulle tvinge brugeren til at køre musen tværs hen over siden gang på gang for at benytte diverse elementer.

Desuden, valgte vi også at lave om på det da vi gerne ville kunne supplere Business Mindset med et mere professionelt udseende på deres platform, der samtidigt er funktionelt ift. deres brugeres behov. Da den primære bruger netop er en herre oppe i alderen, integrerede vi derfor større font størrelser i designets helhed for at sikre en god brugeroplevelse. Sammenlignet ses det lettere hvor stor en forskel det gør for designet at vi ændrede layoutet af elementerne





Desuden valgte vi også at fremhæve diverse elementer for brugeren så det er langt mere intuitivt for eventuelt svagtseende at kunne navigere rundt og benytte platformens forskellige features.



REST METODER

[Forfatter: Jonas]

Vi har fastlagt funktionaliteterne og dermed de metoder vi har brug for, derved har grundlaget som vi bare kan springe ud i og kode vores rest vider.

Det eneste forskellige i GET-metoderne er de queries der bliver eksekveret på database i forhold til det data de skal returnere.

- **GET - memberInTeam**

```
const string sqlstring = "SELECT * FROM dbo.BM_ParticipateMemberInTeam";
```

- **GET - getTeamByName**

```
const string sqlstring = "SELECT * FROM dbo.BM_Team where teamName=@teamName";
```

POST requests er åbenlyst anderledes fra GET, eftersom der nu skal pushes data op i vores database og ind i et table.

Som det ses i koden nedenfor, skal der gives et “ParticipateMemberInTeam” objekt med når metode skal kaldes. De attributes der er inden i selve klasse på REST siden, stemmer overens med de **Navne** og **Værdier** der er i det table hvor data'en ønskes at blive indsat.

```
public void AddMemberToTeam(ParticipateMemberInTeam participateMemberInTeam)
{
    const string Insert =
        "insert into dbo.BM_ParticipateMemberInTeam (teamId, memberId) values (@teamId, @memberId)";

    using (SqlConnection databaseConnection = new SqlConnection(ConnectionString))
    {
        using (SqlCommand insertCommand = new SqlCommand(Insert, databaseConnection))
        {
            databaseConnection.Open();
            insertCommand.Parameters.AddWithValue("@teamId", participateMemberInTeam.teamId);
            insertCommand.Parameters.AddWithValue("@memberId", participateMemberInTeam.memberId);

            insertCommand.ExecuteNonQuery();
        }
    }
}
```

JAVASCRIPT

[Forfatter: Christoffer, Ian & Jonas]

Vi har gennem skoleforløbet tidligere haft arbejdet med JavaScript og tilhørende libraries som f.eks. Node.JS. Dog har vi i vores projektplanlægning, vi har valgt at fravælge brugen af sådanne libraries.

Grunden til det er at vi ikke føler os kompetente nok i brugen af basis JavaScript endnu, til at vi vil springer ud i forskellige libraries - selvom det helt sikker ville være muligt at finde brugbare metoder i dem.

Vores javascript filer indeholder GET og POST requests, hvor vi har benyttet os af Ajax (Asynchronous Javascript And Xml) til at kalde vores requests.

Nedenstående koder er vores GET-base funktion, som alle vores GET-metoder vil kalde, den er nemlig sat op til at request header og opretter et request objekt.

```
/* Get Base Request */
function getAjax(url, success) {
    var xhr = window.XMLHttpRequest ? new XMLHttpRequest() : new ActiveXObject('Microsoft.XMLHTTP');
    xhr.open('GET', url);
    xhr.onreadystatechange = function () {
        if (xhr.readyState > 3 && xhr.status == 200) success(xhr.responseText);
    };
    xhr.setRequestHeader('X-Requested-With', 'XMLHttpRequest');
    xhr.setRequestHeader('Content-Type', 'application/json');
    xhr.send();
    return xhr;
}
```

Her ser hvordan base funktion bliver brugt på vores Create member side, hvor vi bruger den til at adde members til vores listview - som det også ses, giver vi som det eneste, en URL med til vores rest kald.

Udover requesten, opdateres viewet (*en member bliver fysik added til viewet*) og der bliver også kontrolleret om den member der skal addes rent faktisk eksistere i databasen, da et krav fra kunden var, at det ikke skulle være muligt at tilføje en mail der ikke er tilknyttet en personlighedstest.

```
/* Add new member function */
function addMember() {
    getAjax('https://bmrestlibtest20190508033724.azurewebsites.net/Service1.svc/member/' + find_team_members.value, function (data) {
        var memberId = JSON.parse(data);

        //checks if member is exists
        if (!memberId == 0 || !memberId == null) {

            //checks if member is already added to the list
            if (!memberIds.find(element => element === memberId)) {
                var ul = document.getElementById("member_list");
                var li = document.createElement("li");
                li.setAttribute('id', find_team_members.value);
                li.appendChild(document.createTextNode(find_team_members.value));
                ul.appendChild(li);

                /* Clear textfield after member is add*/
                document.getElementById("email_textbox").value = "";
                console.log(memberId);

                // adding member id to the list of member ids
                memberIds.push(memberId);
            } else {
                //TODO some kind of error
                console.log(memberId);
            }
        }
    });
}
```

CORS

[Forfatter: Christoffer]

Da vi skulle til at bruge vores REST med de endpoints vi havde oprettet, stødte vi på et problem som vi ikke havde forudset - på grund af de standarder sikkerhedsprotokoller der er på domains, blev vores XMLHttpRequests stoppet af Cross-origin resource sharing, **CORS**, policy (da vi ikke havde CORS sat op)

CORS er en mekanisme som giver mulighed for at lave requests sikkert mellem to eller flere forskellige domains, med CORS kan du nemlig specificere hvilke fremmede domains der skal kunne tilgå hostens services.

XMLHttpRequests følger som default "same-origin policy", altså at man kun kan lave requests til den samme domain, med mindre den fremmede domain indeholder de rigtige CORS headers, derfor blev vi nødt til at implementere CORS headers på vores azure service.

Dette er meget simpelt gjort på azure, her kan man i API settings, nemt tilføje CORS til servicen. Som vi gjorde på følgende måde:

The screenshot shows the Azure portal interface for managing API settings. The URL in the address bar is [Home > All resources > BMRestLibTest20190508033724 - CORS](#). The main content area is titled "BMRestLibTest20190508033724 - CORS" and "App Service". On the left, there's a sidebar with navigation links: "Easy APIs", "API", "API definition", **CORS** (which is selected and highlighted in blue), "Monitoring", "Alerts", "Metrics", "Diagnostics logs", "Log stream", and "Process explorer". The main pane has tabs for "CORS" and "Request Credentials". The "CORS" tab contains a description of what CORS does: "Cross-Origin Resource Sharing (CORS) allows JavaScript code running in a browser on an external host to interact with your backend. Specify the origins that should be allowed to make cross-origin calls (for example: http://example.com:12345). To allow all, use '*' and remove all other origins from the list. Slashes are not allowed as part of domain or after TLD." Below this, there's a section for "Request Credentials" with a checkbox for "Enable Access-Control-Allow-Credentials". The "Allowed Origins" section has a text input field containing "*" and an ellipsis button "...".

Ved at tilføje stjernen i "Allowed Origins" bliver der oprettet CORS headers til servicen og ved at skrive "*", er det ikke kun et specifikt fremmed domain der kan tilslutte vores service, men derimod alle. Vi kunne også have givet adgang til kun et domain, men da vi ikke har nogle krav om dette har vi bare givet adgang til alle.

FIREBASE & AUTHENTICATION

[Forfatter: Kasper]

Firebase er i vores tilfælde det åbenlyse valg ift. autentifikation af brugere på Business Mindset's platform. Dette lægger sig til dels grundet i at vi førhen har haft erfaring med oprettelse af vores brugerdatabase samt, login gennem Firebase. I Firebase har vi som udgangspunkt gratis adgang til en basis plan der lader os skabe en en database der i vores tilfælde ikke sætter restriktioner der vil forhindre vores udvikling på nuværende tidspunkt. Dette er blandt andet at der maks kan være 100 unikke forbundet enheder til databasen på samme tid, hvilket fordelagtigt passer til skabelse af prototyper, som i vores tilfælde ikke har henblik på høj trafik eller anden belastning.

Der er både fordele som der selvfølgelig er ulemper ved brugen af Firebase. Selvfølgelig er et stort salgspunkt for selve Firebase, dets brugervenlighed. Det er super let i og for sig selv at sætte diverse af deres værktøjer op ift. behov i dit produkt. Så der spares en masse manuelt arbejde og du får hvad du betaler for hvilket i sjældne tilfælde alt efter din sidestruktur kan ende med at blive en dyr fornøjelse. Skulle du mod forventning ud af det blå, få en mængde kæmpe trafik på din side der registrerer sig som brugere. Ender du også med at skulle betale en væsentligt højere regning, nogle brugere har meldt at de pludseligt har skyldt Google flere tusinde dollars på meget kort tid. Foruden dette, arbejder vi med proprietær software. Det er i bund og grund Google der bestemmer hvad der skal ske med din database mm. Udgiver de en opdatering der pludselig øde-lægger noget et andet sted, er det dit eget ansvar at adaptere til dette. Foruden dette er det også en risiko i sig selv at have et andet firma stå til ansvar for din data. Eksempelvis opkøbte Facebook bl.a. firmaet Parse tilbage i 2013. Parse var en platform der tilsvarende Firebase hjalp mobiludviklere med at udvikle deres applikation og sørgede for bevarelse af data mm. online. Blot fire år efter firmaet blev opkøbt, lukkede Facebook det ned i 2017. Google opkøbte i 2014 Firebase og ser umiddelbart ud til at understøtte det en masse, men Google har en meget lang liste over projekter de har lukket ned, så man kan desværre nok ikke vide sig helt sikker.

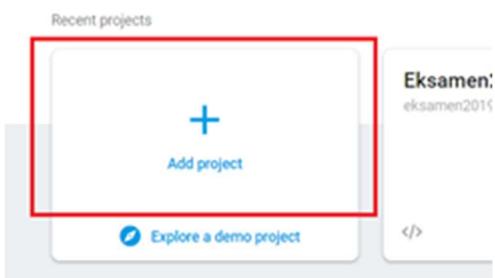
På platformen ønsker vi fra et designperspektiv ikke at give potentielle brugere adgang til vores produkt hvis ikke de først logger ind. Dette er et valg vi har truffet da vi ikke ønsker at personer uden relevans for produktet, ender med at skabe problemer bla. ift. vores skabte teams og deres tilhørende data. Altså har vi valgt at det er nødvendigt at logge ind fra start. Hvad Firebase i grove træk kan tilbyde, er en realtime database der synkroniserer data, der i vores tilfælde er registrerede brugere, i realtid i clouden. Dette vil sige at skulle vi lave ændringer i denne data(ændring af password/e mail eksempelvis) , vil dette blive synkroniseret og relevante enheder/brugere vil blive notificeret omkring disse ændringer, alt sammen nærmest øjeblikkeligt.

Først, skal man selvfølgelig logge ind eller lave sig en Google Konti. Herefter skal som det første egentlige trin, lave et nyt projekt til vores projekt:

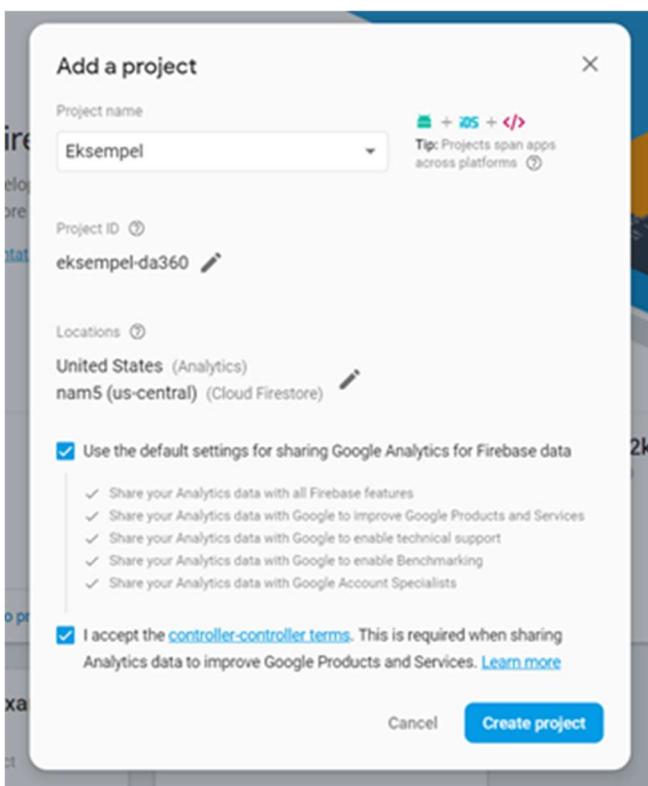
Welcome to Firebase!

Tools from Google for developing great apps, engaging with your users, and earning more through mobile ads.

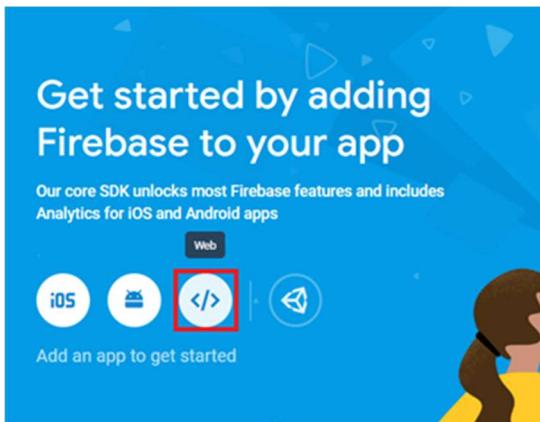
[Learn more](#) [Documentation](#) [Support](#)



Dernæst skal vi simpelt nok bare vælge et navn til projektet:



Med de to første trin ude af verdenen, vælger vi nu hvorvidt at vi udvikler til enten: iOS, Android eller Web. I vores tilfælde er det web:



Som det sidste skridt vi udfører på Firebase side får vi et code snippet som vi skal copy/paste ind i vores projekt for at kunne skabe forbindelse mellem Firebase og vores projekt.

[X Add Firebase to your web app](#)

1 Register app

App nickname: [?](#)

Also set up **Firebase Hosting** for this app. [Learn more](#) [?](#)
Hosting can also be set up later. It's free to get started anytime.

[Register app](#)

2 Add Firebase SDK

Copy and paste these scripts into the bottom of your <body> tag, but before you use any Firebase services:

```
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/6.0.4.firebaseio.js"></script>

<!-- TODO: Add SDKs for Firebase products that you want to use
      https://firebase.google.com/docs/web/setup#config-web-app -->

<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "AIzaSyC00Rj80bumQY8rhjq_mRE2c05suila88",
    authDomain: "eksempel-da360.firebaseioapp.com",
    databaseURL: "https://eksempel-da360.firebaseio.com",
    projectId: "eksempel-da360",
    storageBucket: "eksempel-da360.appspot.com",
    messagingSenderId: "413766492941",
    appId: "1:413766492941:web:86a40580fe711f52"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
</script>
```

Learn more about Firebase for web: [Get Started](#) [?](#), [Web SDK API Reference](#) [?](#), [Samples](#) [?](#)

[Continue to console](#)

Tabellen her er taget fra vores projekt og koden er derfor ikke lig den fra billede foroven da det bare er lavet som et eksempel. Dog er principippet det nøjagtigt samme. Dog har vi tilføjet en ekstra linje i form at script nummer 2 "firebase-auth.js" der sikre os adgang til autentifikationsværktøjet.

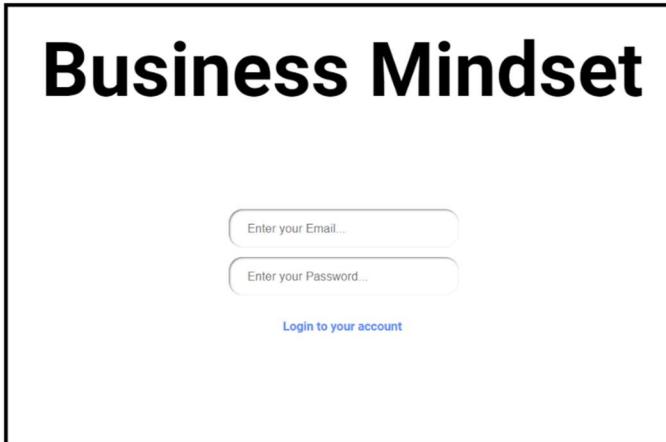
```
<!--FIREBASE START-->
</script>
<script src="https://www.gstatic.com/firebasejs/6.0.2.firebaseio.js"></script>
<script src="https://www.gstatic.com/firebasejs/6.0.2/firebase-auth.js"></script>
```

```

<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "AlzaSyDFNC5YoSqUeC7t_7_inWRBNPptXsHzDqq",
    authDomain: "eksamen2019-1d7a6.firebaseioapp.com",
    databaseURL: "https://eksamen2019-1d7a6.firebaseio.com",
    projectId: "eksamen2019-1d7a6",
    storageBucket: "eksamen2019-1d7a6.appspot.com",
    messagingSenderId: "162724010993",
    appId: "1:162724010993:web:89058f65b637441c"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
</script>
<!--FIREBASE END-->

```

Til sidst er her vores implementering af en loginfeature der tager brug af autentifikationsværktøjet.



Vi lavede en funktion der ved klik på loginknappen udføres. Vi har lavet 3 konstanter i form af: **email, password & auth**. email såvel som password konstanterne, tjener samme formål - nemlig at tage bruger input fra vores input fields i form af en email og et password for at kunne logge på. Derimod er konstanten "auth"s er en implementering af Firebase som vi netop nævnte længere oppe. Formålet med auth, er reelt set at minimere hvor meget vi skal skrive men selve "firebase.auth()" er Firebase værktøjet med alle de funktioner vi ønsker for at kunne skabe en ordentlig autentifikation på siden. **auth()**, har en masse ekstra kald som vi kan kalde på den. Eksempelvis som vi gør det "**signInWithEmailAndPassword**" som man nok let kan gætte er login med e mail og password. Men der kan også kaldes på andre autentifikationsmuligheder såsom Google, Facebook telefonnummer mm.

Desuden catcher vi længere nede i koden exceptions i form af variablen **errorCode**, som har til formål at fortælle brugeren hvis ikke de har indtastet en rigtig email/password kombination.

```

function login() {
  const email = email_field.value;
  const password = password_field.value;
  const auth = firebase.auth();

  auth.signInWithEmailAndPassword(email, password)
    .then(function () {
      console.log("Redirecting to home")
      window.location.href = "home.html"
    })
    .catch(function (error) {
      var errorCode = error.code;

      if (errorCode == 'auth/invalid-email') {
        console.log('Invalid Email')
        window.alert('Invalid Email')
      }
    })
}

```

```

} else if (errorCode == 'auth/wrong-password') {
    console.log('Wrong Password')
    window.alert('Wrong Password')
} else {
    console.log('Error: ' + errorCode);
    window.alert('Error: ' + errorCode);
}
});
}

```

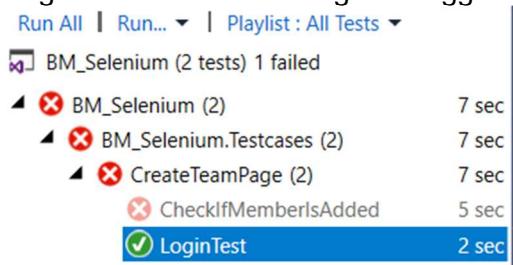
TESTS

[Forfatter: Ian]

Selenium

Eftersom vi færdiggjorde vores user story fra første sprint i sprint #2, ændrede det ikke noget på testen fra sprint #1. Som tidligere nævnt havde Business Mindset prioriteret en loginfeature som vi fik implementeret, hvilket vi nåede indenfor sprintet. Dette betyder at vi har fået implementeret mere GUI og funktioner der skal testes igennem. Vores nuværende selenium test projekt er ikke så omfattende, men ideen med at vi kan tilføje flere test cases baseret på nye funktioner i programmet gør at vores test projekt giver værdi i den lange ende.

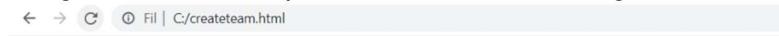
Vores login side indeholder de grundlæggende 3 felter: Input til email og kode samt en log ind



knap.

Vores login kode skal givet en gyldig email og kode kunne logge ind til hovedmenuen, hvorefter man skal kunne navigere til create team siden, så vores tidligere test case fra sprint også fungerer. Men når visual studio bliver kørt, får vi følgende resultat i test projektet. Dette betyder at vores tidligere test case som førhen virkede, nu ikke består. (se billedet til højre)

Følger vi den automatiske test script ender vi på følgende side, vi kan se at vores create team eksisterer ikke, ellers skulle den være blevet flyttet. Herfra kunne vi konkludere at der var sket en ændring i vores Javascript siden vores sidste merge.



Din fil blev ikke fundet

Den kan være blevet flyttet eller slettet.

ERR_FILE_NOT_FOUND

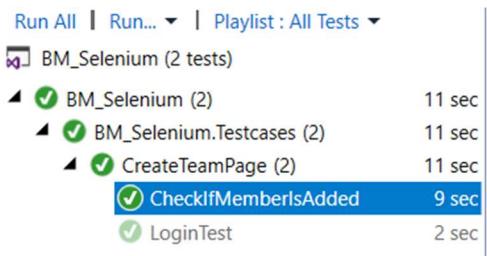
Grunden ligger i, at vi bruger javascript til at tage den indtastet string og navigere direkte til stringen som endpoint i stedet for hele URL, som peger på HTML siden i forhold til hvor den ligger inde i projektet. (Markeret med rødt og understreg i koden nedenfor)

```
</div>
<!--Div for nav buttons, when you're logged in-->
<div id="button_div" class="loggedin_btns_div">
<button id="btnCreateCompany">Create Company</button>
<button id="btnCreateTeam" onclick="javascript:location.href='/createteam.html'">Create Team</button>
<button id="btnTeamResults">Team Results</button>
</div>
```

Løsningen var at benytte "window.location" som gør onClick kigger i hele projektet efter en fil med tilsvarende navn og navigere til denne.

```
</div>
<!--Div for nav buttons, when you're logged in-->
<div id="button_div" class="loggedin_btns_div">
<button id="btnCreateCompany">Create Company</button>
<button id="btnCreateTeam" onclick="window.location='createteam.html'">Create Team</button>
<button id="btnTeamResults">Team Results</button>
</div>
```

Efter vi fik ordnet vores lille fejl ser vores følgende test case således ud:



Eftersom login test casen bliver godkendt af Selenium, kan den indføres i [Setup] metoden da det er første skridt for at benytte sig af systemet. (Se nedenstående kode)

Dette betyder at Setup vil blive kørt hver gang man starter en test case, derefter kan man navigere til de forskellige sider. Fordi vores login test er placeret i Setup metoden vil den ikke ses som en testcase længere i visual studio da den ikke er markeret med "[TEST]".

```
[SetUp]
public void Setup()
{
    driver.Navigate().GoToUrl(Helper.pageUrl() + "/index.html");
    //driver.Navigate().GoToUrl("http://127.0.0.1:5500/");

    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(5000));
    IWebElement emailfield = driver.FindElement(By.Id("email_field"));

    // wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementToBeClickable(By.Id("btnLogin")));

    // types in email
    emailfield.SendKeys("k@k.dk");
    // types in password
    driver.FindElement(By.Id("password_field")).SendKeys("123123123");
    // click on login
    driver.FindElement(By.Id("btnLogin")).Click();
}
```

INFORMATIVE WORKSPACE

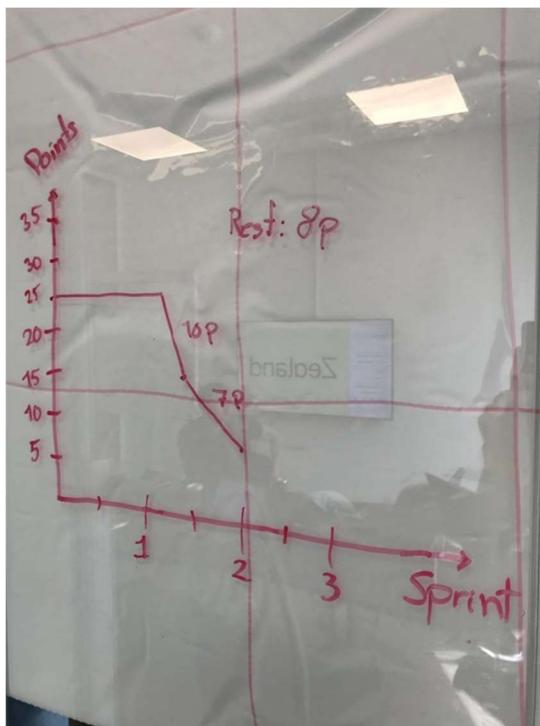
[Forfatter: Kasper, Ian, Christoffer & Jonas]

Burndown Chart

Vi har indtil videre ikke haft udbytte af burndown chart og der har heller ikke været brugt specielt meget tid på at opdatere og evaluere burndown chartet - grunden er selvfølgelig at vi ikke har færdiggjort så mange use cases og vi har dermed ikke optjent nogle storypoints som kan tælle ned på vores burndown chart.

Der er i alt 35 story points til rådighed ud igennem vores projekt, dog er det ikke realistisk at vi kan nå de 35 point, da der også skal sættes tid af til at skrive rapport i den sidste ende af projektet og i starten havde vi presprint som heller ikke gav mening at tælle med, derfor er det realistisk at vi kan nå 25 story point i vores projekt. Burndown chartet starter derfor på 25 og tæller ned.

Hvis man ser på den efterfølgende figur, ser man som allerede beskrevet, vores flade start hvor der ikke blev færdiggjort nogen usecases. Grunden til dette er vores store use case, som vi havde i sprint et, det blev ændret i sprint to's retrospective hvor vi inddrog et Kanban Board, til netop at opdele vores use cases i mindre tasks.



Dette resulterede i at vores Burndown Chart oftere kunne opdateres og dermed skabe mere transparent i teamet, så alle medlemmer også kan holdes opdateret.

Velocity

Selvom vores Burndown Chart ser meget kantet ud, så har vi stadig holdt vores forventede velocity. Som tidligere nævnt havde vi 25 story points der blev fordelet på 3 sprints, (8.3 story points per sprint) Og nu har vi et sprint tilbage og ifølge vores Burndown Chart er vores resterende points = 8 points, som der kan medtages uses cases for i sprint #3. Dermed tyder det på at vi har været gode til at estimere, nu hvor vores velocity står til at gå i mål meget tæt på sprint 3's afslutning.

OPSUMMERING AF SPRINT #2

[Forfatter: Kasper & Christoffer]

Kontakt med business mindset evaluering #2

Vores product review blev rykket til om mandagen da Andrew var udrejst på forretningsmøde og var derfor ikke til at træffe - dette resulterede også i at vi kun havde et enkelt 20min brief.

Udsættelsen af mødet samt af sprint reviewet, havde ikke nogen reel effekt på processen, da kravene var specifikke nok og vi havde at materiale der skulle bruges (*alt vores dummi data*)

Hvilket stадie er produktet i pt?

Det er muligt at logge ind hvis man har fået en adgangsgivende bruger, herfra er det så muligt at navigere til 1 af 3 muligheder (*see result, create team og edit team*), *det eneste side der kan nавигeres til er "create team"* - her er det muligt at tilføje medlemmer til et listview hvor hvert medlem vil blive tjekket om vedkommende er i databasen for mulige medlemmer (*brugere med en complete brugertest*).

Derudover kan teamet navngives og alt det information der bliver indtastet kan gemmes og hentes i databasen.

Hvad er planen fremadrettet?

Afhængig af om der kommer kravændringer frem ved sprint reviewet, så skal sprint #3 planlægges og dette sprint er som tidligere nævnt kun et halvt sprint, den resterede tid skal bruge på at skrive det sidste på rapporten, opdatere billeder og indsætte kode blokke.

Vi skal også have planlangt et sidste skype-møde hvor vi med John og Andrew skal snakke om de fortsat vil have os til at videreudvikle på produktet samt at give en evaluering af samarbejdet set fra deres side og hvad de føler de har fået ud af et samarbejde med os og hvad der kunne gøres bedre.

SPRINT REVIEW

[Forfatter: Ian & Jonas]

Sprint review ligger sidst i sprintet (en fredag) hvor Andrew og John ville blive fremvist det kørende produkt, men eftersom Andrew var beskæftiget om fredagen d. 24-05-19, besluttede vi i fællesskab at rykke mødet til mandag d. 27-05-2019 - Det er i starten af sprint #3, hvilket betyder at der fra om morgen indtil vi har haft mødet ikke kan lave andet en at arbejde på rapporten, da det ikke ville give mening at gå videre uden en bekræftelse.

Det er kom ud af fremvisning var:

- Bekræftelse af vores arbejde (*loginfeaturen, main menu, og nyt design*), Andrew var godt tilfreds med hvordan det virkede og så ud.
- **Requirements liste** - Vi snakkede lidt om hvordan vi ville have vores opret team side til tjekke (validere) de email der blev tastet ind, da det ikke skal være muligt at tilføje en email der ikke er tilknyttet en bruger som har taget en test.
Andrew sagde at vi måtte samle en liste med API kald som vi fandt nødvendige for at få vores produkt til at virke optimalt.

Req liste:

- GET - det skal være muligt at få en brugers oplysninger pga. en email (*så vi kan tjekke at mailen er til stede i databasen og samtidig få brugerens navn, som skal vises sammen med mailen i listviewet efter tilføjelsen af en member*)
- GET - *Det skal også være muligt at få det endelige resultat fra en brugers personlighedstest. (altså de forskellige kategoriers score)*
- **Kravændringer** - Andrew havde også nye krav som han synes vægtede højere end de nuværende use cases i vores product backlog, så vi står endnu engang i en situation hvor vi har fået tilføjet nye krav.
Det betyder ligesom sidst at de skal opskrives som use cases, estimeres og smides i product backloggen, afhængig af deres prioritering.
 - As a admin I want to be able while creating a team to perform **UD** (*Update & Delete*) in **CRUD operations** on added members, so that I can delete and rename members.
 - As a admin I want to be able to have a separate page, where I can manage all created teams, so that I can:
 - See all created teams
 - Delete existing teams
 - Rename existing teams
 - See all members within a existing team
 - Delete members from existing teams
 - Add members to existing teams
- **Validering af information** - Andrew så gerne at vi implementerede noget logik der validerede informationer der blev indtastes
 - E-mail, skal valideres ift. om den er til stede i databasen
 - Teamname skal tjekkes for input (*der man ikke skal oprette et navnløst team*)
 - Listviewet skal tjekkes ift. om der er tilføjet members, da det er et krav at der skal være minimum ét member i et team for at kunne oprette en team.

- **Ros** - Endnu engang modtog vi ros fra måden vi har arbejdet på og ikke mindst vores produkt vi har leveret. (Selvom det ikke er det største produkt vi har udviklet)
Grunden er som Andrew forklarede, at vi kommer med et helt nyt syn på ideer og tanker til funktionalitet og design, som uanset produktets stadie bidrager positivt til Business Mindset

OBS! ÆNDRING AF PROBLEMFORMULERING

[Forfatter: Ian, Jonas, Christoffer & Kasper]

Da vi snakkede med John og Andrew første gang i Belfast omkring et samarbejde, var formålet at fremvises disse metrikker og gøre dem visuelle og benytte deres logik til at identificere om et team er ideelt for at opnå succes.

Men grundet en kombination af mange krav ændringkravændringer, John manglede at færdiggøre logikken for hvordan disse score skulle vægtes samt manglende REST funktioner til at trække data ud fra deres database, er vores problemformulering ændret sig. .

Den gamle:

- **Hvordan udvikler vi en ny funktion, der på baggrund af en personlighedstest kan identificere om et team er ideelt for at opnå succes.**

Den nye:

- **Hvordan kan man designe og udvikle et fundament, der har til formål at give Business Mindset en start platform der kan bruges til visualisering af succes metriker**

Opfyldelse af denne problemformulering ville sætte Business Mindset i en position, hvor de bare skulle tilføje deres udarbejde logik og arbejde med visualisering af denne og så er deres funktion i hus.

Sprint #3

SPRINT RETROSPECTIVE

[Forfatter: Ian, Jonas, Christoffer & Kasper]

Som fortsat benytter vi os af sprint retrospective hvor vi reflekterer over hvad vi kunne gøre bedre, hvad vi har gjort godt og hvad vi skal stoppe med at gøre. Vores retrospective fandt sted mandag d. 27-05-2019 og vi blev enige om følgende:

Keep:

Dagens motto

Dagens motto helt sikkert bidraget til stemningen i gruppen, da der til tider har stået nogle sjove/skøre mottos på tavlen og er med til at give lidt pusterum, når det er nødvendigt.

Informative Workspace

Vores informative workspace bryder vores kedelige og grå arbejdssted med relevante projektinformation. Efter sprint #1 indså vi at vores user stories kan være ret omfattende og derfor virker det uoverskueligt med hvilke tasks der er færdige.

Derfor valgte vi at indføre **Kanban Boardet**, hvilket har givet teamet det udbytte som vi havde håbet på. Det har givet os et dagligt overblik over processen, hvad der mangler og hvad vi er færdige med. Visuelt er det meget nemmere at danne et overblik over eftersom vi har opdelt tasks i forskellige kategorier (GUI, funktionalitet, error box og firebase). Eftersom vi havde stor succes med dette i sprint #2 har vi selvfølgelig valgt at medtage det i de fremtidige sprints

Stop

Dummy data

Eftersom vi har fået implementeret vores dummy data ind i vores system så funktionaliteterne virker, er næste skridt at tage Business Mindset rigtige data ind i stedet. Derfor ser vi at dummy data ikke længere har en funktion for dette sprint. Dette skal dog ikke forveksles med test data, da vi stadig har brug for dummy data til at teste funktionaliteten.

2 x 20 min. homework

Målet med dette tiltag var at indhente rapport sider, men eftersom vores produktivitet i skolen har vist sig være været bedre end forventet har vi valgt at sløjfe denne løsning. En anden faktor for at stoppe med at benytte os af denne løsning er vores selvdisciplin har været minimal, da vi oftest har meget at se til i hverdagene efter skole og derfor bliver homework prioriteret til sidst.

Do

Daglige rapport mål

I forlængelse med afskaffelsen af homework valgte vi at indføre rapport mål i vores informative workspace, så vi hver dag opdaterer hvor mange tegn rapport indeholder på nuværende tidspunkt. Dette bliver regnet om til sider, så vi ved hvor mange sider vi som minimum skal skrive for dagen, for at kunne nå målet. Dette tiltag har indtil videre givet os et overblik over rapportens situation og hvad der forventes af os for at kunne færdiggøre en veldokumenteret rapport.

Mangler:	41396 Tegn
Dagens Mål:	5000 Tegn
Projektets Samlede Mål:	136000 Tegn

SPRINT PLANNING

[Forfatter: Christoffer, Ian, Kasper & Jonas]

Som tidligere nævnt fik vi to nye use cases som først og fremmest skal spilles planning poker om for at få dem estimeret og dernæst skal product backloggen opdateres. Resultatet af planning poker var som følgende:

- Use case 1: As an admin I want to be able while creating a team to perform **UD.... - 4 points**
- Use case 2: As an admin I want to be able to have a separate page, where I can manage all created teams... - **6 points**

Pointene er givet på baggrund af vores nylige tilegnet erfaring fra opsættelse af **CR**, som blev estimeret til 10 story points - dette har fået os til at mene, at vi kan implementere **UD** hurtigere.

Derudover er sprintet også kun 5 dage langt, derfor har vi valgt at medtage **use case 1**, da den passer perfekt i forhold til disponibele tid.

Samtidig vil vi have fokus på alt de validering andrew ønskede der skulle ske og nogle små rettelser i koden (*redundans*).

The screenshot shows a digital workspace interface for managing a project backlog. The interface is divided into several sections:

- Backlog:** A list of user stories with their descriptions and estimated points (e.g., 6 points). One story is highlighted with a red background.
- Sprint 3 - 27-05-2019 / 07-06-2019:** A list of user stories with their descriptions and estimated points (e.g., 4 points). One story is highlighted with a red background.
- In progress:** A list of user stories with their descriptions and estimated points (e.g., 0/2).
- Ready for review:** A list of user stories with their descriptions and estimated points (e.g., 0/2).
- Sprint 2 Done:** A list of user stories with their descriptions and estimated points (e.g., 7 points). One story is highlighted with a green background.
- Sprint 1 Done:** A list of user stories with their descriptions and estimated points (e.g., 10 points). One story is highlighted with a green background.

Each user story card includes a "Add a card" button and a "Show More" button.

HVORDAN UDVIKLEDE VI

[Forfatter: Christoffer, Ian, Kasper & Jonas]

AddMember metode er blevet opdateret, så hvert element der bliver tilføjet til listview vil have en onclick funktion, der kan returnere det index.

```
refreshArray(); //updates the view and give all ele-  
ments  
.  
accordingly index values  
  
console.log(memberId); //logs the id  
  
liNode.onclick = function(){  
    index = tab.indexOf(liNode.innerHTML);  
    console.log(liNode.innerHTML + " INDEX = " + index)  
    find_team_members.value = liNode.innerHTML; //set selected value to the emailfield.  
};
```

Alle de elementer der er i viewet, har et unik index i en intern liste, som bliver returneret når et element bliver selected, som gør os i stand til at slette elementer ud fra hvad der er selected.

```
var tab = [], index;  
var items = document.querySelectorAll("#member_list li");  
  
function refreshArray(){  
    tab.length = 0; // reset the counter for the for-loop  
    var items = document.querySelectorAll("#member_list li"); //gets all the li elements there's present within the list  
  
    for(var i = 0; i < items.length; i++){ //put elements into the list.  
        tab.push(items[i].innerHTML);  
    }  
}
```

PROBLEMSTILLING

Der er dog opstået et problem som ikke var lige til at løse indenfor den mængde tid som vi havde til rådighed i sprintet.

Som det er lige nu, bliver elementer i listviewet tildelt et unikt id (*altså deres index placering i viewet*) som bliver brugt til at select elementer med.

Derudover bliver hver indtastet e-mail valideret i forhold til databasen, hvis e-mailen eksisterer, vil dens unikke id blive returneret. Vi har altså to lister hvor der skal slettes id'er fra som ikke bare lige kan gøres, fordi værdier ikke er pairet op.

Hvordan skulle vi fx kunne slette et valgt element (jonasmanley@hotmail.com) på index 0 i en listen og med et unikt id i databasen = 4 i en anden liste, der er totalt uafhængige af hinanden?

Hvis vi skulle løse problemet, skulle vi have benyttet os af **dictionary** hvor vi har Value/key pairs, så kunne vi nemt slette pga. værdien der returneres fra database, da den værdi er koblet til et specifikt index i viewet.

OBS! Det er selvfølgelig noget vi vil fikse, hvis vi indgår i videregående samarbejde med Andrew og John.

Error og success beskeder i viewet

En ting Andrew lagde vægt på i sprint review for sprint #2 var at der blev kommunikeret fejlbesked til brugeren, da den eneste mulighed for at se om noget gik godt eller dårligt pt, er ved at kigge i loggen (*hvilket er langt fra brugervenligt*).

Create Team

CHOOSE A TEAM NAME AND INVITE MEMBERS

Give the team a name and add members to the team

Choose Team Name... Insert Email of member... Add Member

Create Team

CHOOSE A TEAM NAME AND INVITE MEMBERS

Team is created

CodeSharks Insert Email of Member... Add Member

Jonasmanley@hotmail.com
chris31@live.dk
ianmfrost0@gmail.com
kasperkofoed@live.dk

Create Team

```
if(idCheck != 0){  
    document.getElementById('isa_error').style.display = "flex";  
    const error_message = document.getElementById('error_message');  
    error_message.innerHTML = "Team name is already taken";  
}  
  
if(document.getElementById('name').value == "" && tab.length == 0){  
    document.getElementById('isa_error').style.display = "flex";  
    const error_message = document.getElementById('error_message');  
    error_message.innerHTML = "Give the team a name and add members to the team";  
}  
else if(tab.length == 0 || tab == null){  
    document.getElementById('isa_error').style.display = "flex";  
    const error_message = document.getElementById('error_message');  
    error_message.innerHTML = "Please add members to proceed to create a team";  
}  
else if(document.getElementById('name').value == ""){  
    document.getElementById('isa_error').style.display = "flex";  
    const error_message = document.getElementById('error_message');  
    error_message.innerHTML = "Please give the team a name";  
}  
else if(tab.length > 0 && document.getElementById('name').value == ""){  
    document.getElementById('isa_error').style.display = "flex";  
    const error_message = document.getElementById('error_message');  
    error_message.innerHTML = "Please give the team a name";  
}
```

Ovenstående kode viser de fem forskellige scenarier af fejludfyldt view, som alle vil udløse en relateret fejlbesked til at blive vist i viewet.

Hvilket var ret simple at udarbejde: Giv så lidt information som overhovedet muligt og prøv at udfører en given funktion og skriv hvad der forventes at ske.

Derefter arbejdes der stille og roligt mere og mere information ind i viewet indtil det hele er udfyldt og alle scenarier er fundet.

En bruger prøver at oprette et team i følgende scenarier:

- Der er ingenting indtastet og brugeren prøver at oprette et team - *Fejlbesked*.

- Der er indtastet et team navn og ingen members er tilføjet - *Fejlbesked*.
- Der er indtastet mindst en member men intet team navn - *Fejlbesked*.
- Der er indtastet mindst en member og et allerede eksisterende team navn - *Fejlbesked*.
- Der er indtastet et team navn og mindst en member - *Success Besked*.

Allerede eksisterende e-mail

E-mails bliver også valideret i forhold til om de allerede er blevet tilføjet til viewet. (fra sprint #2)

Create Team

CHOOSE A TEAM NAME AND INVITE MEMBERS

 Member has already been added

Codesharks

Jonasmanley@hotmail...

Add Member

Jonasmanley@hotmail.com

Create Team

Font Awesome Ikoner

Er et ikon/figur baseret toolkit hvor man få adgang 1.515 gratis ikoner, som man gerne må bruge i kommersielt brug, hvilket er et must ift. vores produkt.

Udover at være gratis, så er Font awesome også super let at implementere i sin code.

1. Man finder det ikon man vil have inde på Font awesome hjemmeside
2. Dernæst indsættes der en generel reference på den html side man ønsker at indsætte sit udvalgte ikon: [<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css" integrity="sha384-fnmOCqbTlWlj8LyTjo7mOUS-tjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr" crossorigin="anonymous">](https://use.fontawesome.com/releases/v5.7.1/css/all.css)
3. Til sidst copy og pastes ikonets reference ind i koder, der hvor ikonet skal vises:

```
<div class="isa_error" id="isa_error">
  <i class="fa fa-times-circle"></i>
  <p id="error_message"></p>
</div>
```

Nu vil der altså inde i vores div "isa_error" være et ikon.

TEST

[Forfatter: Ian & Jonas]

I forlængelse af vores implementering af validering har vi nu fem nye test cases der skal testes.

```
IWebElement succes_element = driver.FindElement(By.Id("isa_error"));

// checks if succesmessage appear
string ActualErrorMessage = succes_element.Text;
string expectedValue = "Please add members to proceed to create a team";

Assert.AreEqual(expectedValue, ActualErrorMessage);
```

Ovenstående kode er det essentielle i én af de nye test cases (*logikken er den samme i alle fem test cases, men fejlbeskederne og indtastet data til viewet ift. fodnote 18*).

Error besked fremprovokeres, hvorefter den bliver grebet fat i (som er en string) og dernæst sammenlignes den værdi som burde fremprovokeret.

Resultat

Som billede til højre bekræfter alt validering der er implementeret på vores hjemmeside, fungerer hensigtsmæssigt.

The screenshot shows a test execution interface with the following details:

- Run All | Run... ▾ | Playlist : All Tests ▾
- BM_Selenium (6 tests)
- BM_Selenium (6) - 39 sec (Passed)
- BM_Selenium.Testcases (6) - 39 sec (Passed)
- CreateTeamPage (6) - 39 sec (Passed)
 - CheckExistingTeam - 8 sec (Passed)
 - CheckIfMemberIsAdded - 3 sec (Passed)
 - CheckMissingMember - 6 sec (Passed)
 - CheckMissingTeamName - 7 sec (Passed)
 - CheckNothingTypedError - 6 sec (Passed)
 - CheckSuccesAlert - 8 sec (Passed)

PRODUCT REVIEW

[Forfatter: Christoffer, Ian, Kasper & Jonas]

Teknisk set har vores sprint #3 kun været en uge lang, hvilket har betydet at vi havde product review fredag d. 31/05 med Andrew. Her fremviste vi vores validering implementation samt hvad vi havde fået på delete funktionen (*vi havde som tidligere nævnt en problemstilling med delete funktionen*). Andrew var dog alligevel tilfreds med hvad vi havde fået taget i betragtning at vi kun havde én uge til at programmere.

Vi forklarede Andrew at vi havde mulighed for at arbejde lidt videre på projektet inden vores fremlæggelse/forsvar af hovedopgaven. Derfor snakkede vi om planen fremadrettet, som var følgende:

- Prioritering af færdiggørelse af delete funktion.
- Påbegyndelse af create company (*POST*), så man kan oprette et company som kan indeholde flere teams
- Der skal også udvikles et redigerings page, som er en side hvor man kan vælge et company og får så muligheden for at vælge et team man gerne:
 - Tilføje eller slette medlemmer
 - Slette et helt team
 - Redigere et team navn

Til sidst snakkede vi lidt om hvor han syntes om projektforløbet, men vi ville egentlig også have det på skrift, så vi kunne vedlægge det rapporten til Censor og vejleder (*eftersom de ikke har haft mulighed for at træffe Andrew eller John*)

Hi Jonas,

More than happy to :) I never got to know any of the backstreet boys well :) something we must remedy in the future.

Overall:

- Great work. You communicated regularly and did so with clarity when updating myself on the project.
- The client was engaged and found your commitment to challenge very impressive. You worked consistently through the project providing a consistent momentum.
- Your team jelled really well. You each contributed using your existing skills very well and this presented very positively to the client.

Well done!

On constructive feedback:

- The limitations of an academic project can prevent certain normal interactions that get involved in normally commercial work.
- Because of lack of budgeting discussions etc resource management conversations were limited.
- Lack of a graphical design process up front meant there was a lack of understanding of what might be delivered the other end of the project.

Hope this is useful for your lecturers.

You guys are all amazing. Especially yourself! It's the structure of courses can hold you back from trying and testing certain processes.

Happy to be interviewed on the same. Or give a testimonial!

Kindest regards
Andrew

SPRINT RETROSPECTIVE OVER ARBEJDSPROCESSEN

[Forfatter: Jonas, Ian, Christoffer og Kasper]

Da tredje sprint er vores sidste sprint, giver det ikke nødvendigvis mening at lave et ordinært sprint retrospective. Dog føler vi at det er en god måde afslutningsvis at reflektere over vores arbejdsprocess for eventuelt at kunne finjustere processen. Jo flere sprints vi potentielt kan kigge tilbage på, desto bedre indsigt får vi både i vores generelle samarbejde, men også så vi enkeltvis ved hvordan vi hver især får mest ud af fremtidige gruppe projekter på videregående uddannelser eller endda arbejdspladsen. Endvidere, da vi fremadrettet stadig ikke ved om hvorvidt at samarbejdet med Business Mindset potentielt kunne fortsætte efter projektets afslutning, er dette et uvurderligt værktøj at have i baglommen.

Der vil være procesværktøjer fra tidligere sprint retrospectives der kommer til at gå igen i dette retrospective. Herunder: **Daily Scrum**, **Sit Together** og **Informative Workspace**. De går igen af samme årsager som i tidligere retrospectives, men vi vælger ikke at gentage det her, da vi ikke ser dette som et rigtigt retrospective så at sige. Dog er der stadig nogle ting vi føler ikke har bidraget positivt/positivt nok til den overordnede proces som vi gerne vil stoppe eller gøre bedre fremadrettet.

Stop:

Eftersom vores projekt lakker mod enden, er det også ensbetydende med at vores rapport bliver færdiggjort. Derfor er **daglige rapport mål** et overflødigt element i vores informative workspace.

Do:

Vi er ikke helt tilfredse med vores brug kanban boardet, selvom vi har haft vores arbejdsopgaver visualiseret i vores informative workspace og at det helt sikkert givet et klart bedre overblik over projektet som helhed, så har det ikke været nemt og overskueligt at se hvem der arbejder på hvad. Vi har ikke formået at bruge kanban boardet ift. de enkelte personer i teamet - som har resulteret i at vi ikke har kunne have et godt overblik over hvem der på et givent tidspunkt arbejder på de enkelte use cases.

Derfor ville løsningen fremadrettet være at skrive sine initialer i bunden, af den use cases som man arbejder på. Hvis der er flere i teamet som arbejder på den samme task, skal begge selvfølgelig stå på.

REFLEKSION

[Forfatter: Christoffer, Ian, Kasper & Jonas]

Afslutningsvis tager vores sidste opgave på studiet sig form af det virkelige arbejdsliv i tech industrien. Fra at have haft siddet på skolebænken til at vi nu skal kunne vise at vi er klar til at forlade den efter 2 års arbejde. Kan det roligt siges at have været en stor men vel mødt mundfuld. Fra 1 - 4 semester hvor vi har haft faste rammer i forhold til hvad det er vi skal lave i undervisningen, til her til sidst på 5 semester, hvor vi ikke har haft normal undervisning i et helt semester. Men tværtimod har haft fået lov til at bruge al den viden vi har tilegnet os gennem skoleforløbet i praksis og virkelig fået en smag for hvad det vil sige at være udviklere.

Vi har arbejdet i samarbejde med Business Mindset, som vi stiftede kendskab til, gennem vores praktikforløb tidligere i semesteret. Holdet bagved er to entreprenører med adskillige andre succesfulde projekter og dette har jo som nævnt resulteret i at vi som studerende arbejdskraft på et sideprojekt har haft lavere prioritet end andre projekter som set fra deres side giver dem mere værdi (Business mindset, er et "hobby" projekt der er udviklet sig og kan blive seriøst.

Problematikken opstår dog i det faktum at selve arbejdsopgaven i sin tid måske har været lidt underurderet i forhold til de kompetencer vi havde stillet os selv til rådighed med.

Det er teknisk set et nyt emne for mange af os, da vi har ikke arbejdet meget med web development før dette projekt, som også har vist sig i de problemstillinger vi har stødt på undervejs.

Der er derfor gået meget tid med at researche og prøve forskellige løsningsmuligheder af. Det har altså af netop samme årsag også været svært at estimere forskellige arbejdsopgavers omfang i forhold til planlægningen og struktureringen, men det har dog tydet på at vores intuition har været meget præcis.

I forlængelse med ovenstående har det været en væsentlig faktor (*som vi også har debatteret et par gange ud igennem rapporten*) at vores kunde er bosat langt væk - som endte ud i at vi brugte lang tid på at komme i kontakt med dem, for at få adgang til deres server, eller når vi skulle安排re møder, som også kunne tage en del mails over flere dage at få på plads. Til dette kunne vi have gjort os mere brug af vores design sprint, som vi kun fik benyttet i pre sprintet. Netop fordi vi med dette værktøj hurtigt kan få tilrettelagt de konkrete krav i stedet for at arbejde ud fra hvad vi antager er rigtigt, mens vi venter på det næste møde, bliver stablet på benene.

Efter vores introduktionsmøde var vores forventninger høje, da projektet lød spændende og som en god mulighed for at få en professionel korrespondance med to veletablerede entreprenører og derved kunne få et indblik i erhvervslivet. Men desværre kom dette ikke helt til udtryk som vi havde forventet - Projektet var lovet som et projekt hvor vi "bare" skulle consume en REST, putte noget data i en database og tilføje noget logik som blev udarbejdet af dem (*vi skulle altså bare kode*).

På trods af snakken om hvordan at vi blandt andet har haft følt at der har været mangel på kommunikation fra kundens side af, har dette også set i bakspejlet været langt bedre for os end hvis projektet havde kørt som smurt hele vejen. Det er gennem fejltagelse og mangler vi lærer at tilpasses os ikke den anden vej rundt. Så alt i alt har dette været en god måde at til dels at afslutte uddannelsen, såvel som det har været en god måde at få en ordentlig fornemmelse af hvordan kundekontakten i sidste ende reelt set former hele processen. Når vi på tidligere semestre har leget kunde for os selv, har vi i bund og grund lidt tilpasset blandt andet kravændringer til hvordan vi nu lige følte de skulle gavne vores projekt bedst muligt. Men dette er bare ikke en realitet i det virkelige arbejdsliv, og derfor har dette også været en uvurderlig lektion for os at kunne tage med videre.

Men med dette sagt, føler vi stadig at resultatet af hvad vi har lavet, viser at vi som i jævnfør af læringsmålene har bevist at vi metodisk kan afdække de krav der er sat til vores produkt og trods

forhindringerne vi er stødt på. I et acceptabelt omfang kan realisere disse krav indenfor givne rammer. Desuden har vi anvendt nye programmeringsteknikker og værktøjer til at få skabt dette produkt og har undervejs kunne dokumentere lige netop hvorfor det er vi har truffet de valg, vi har taget.

Gruppearbejdet har været ligesom i vores andre projekter været: sjovt, udfordrende og fuld af diskussioner der har faglig relevans - nogle gange har vi blevet nødt til at sætte en tidsbegrænsning (*ordstyreren Ian*) på disse diskussioner, da de ofte blev temmelig omfattende.

PERSONLIG REFLEKSION

Jonas - Det største udbytte fra projektforløbet kommer helt sikkert fra, at vi rent faktisk har haft en rigtig kunde (*ikke som tidligere projekter, hvor vi selv har været kunden og skulle komme med krav*), vi har haft skulle indsamle krav for at finde ud af hvad programmet rent faktisk skulle kunne, forstå kravene dem bedst muligt og så selvfølgelig fører dem ud i praksis - i en "rigtig kunde" ligger også uforudsigeligheden, vi har mange gange ud igennem projektet set hvordan kravene pludselig ændre sig, som gjorde at vi har måtte tilpasse os.

Det har også været sjovt at lave et produkt helt fra bunden og få opsummeret hvordan det nu var at man lavede en database, hvordan man kunne hente data via en REST web service og bare i det hele taget, hvordan man planlægger et projekt.

Der har også været mange "argh-ha-oplevelser", som har været hvor vi har siddet en hel dag men hvad der føltes som et simpelt problem, men uanset hvad vi gjorde/prøvede virkede det bare ikke, men i slutningen af dagen eller dagen efter, ville vi have knækket det.

Ian - Vores hovedopgave har givet os mulighed for at benytte vores forskellige kompetencer vi har tilegnet os under praktikperioden. Vi har rykket os som individer og det ses tydeligt under vores arbejdssproces, vi har fået større gåpåmod og vores tilgang til at løse problemer er blevet mere selvstændigt.

Det har været et spændende forløb, som i sidste ende viser hvor meget vi har rykket os fra da vi startede på uddannelsen. Projektet har samtidig givet os mulighed for at benytte os af alt den viden vi har lært igennem hele uddannelsen, det har derfor også været rart at få det genopfrisket.

Vi har igennem uddannelsen ageret som vores egen kunde når vi har skulle skrive projektopgaver, i og med at vi har en reel kunde i et andet land synes jeg at vi har taget opgaven mere seriøst og virkelig ydet vores bedste for opnå den bedste tilfredshed hos kunden.

Især har det været interessant at kunden ikke vidste hvad produktet skulle ende ud i, men at vi løbende gjorde hinanden klogere på det ønskede produkt. Det har også betydet at vores kunde har ændret krav, som vi skulle tage stilling til at prioritere, hvilket man må sige er meget relevant når det kommer til den virkelige verden.

Projektet har givet mig et mere realistisk billede af hvor vigtig kommunikationen er mellem kunde og udviklere.

Christoffer - igennem dette projektforløb har jeg udviklet mine færdigheder i at kommunikere med en kunde og udvikle videre på hans ideer, som til sidst resulterte i et produkt. Jeg synes dette med at finde frem til et produkt sammen med en kunde, har været en spændende men svær opgave. Derfor har det givet mig en forståelse for, hvor værdifuldt det kan være at have nogle gode redskaber til at finde frem til kundens ønsker.

Vi startede med en samtale med kunden, hvor vi fandt frem til hvilke usecases der skulle udvikles. Derefter lavede vi et design sprint, som vi efterfølgende kunne vise til kunden og jo længere inde i forløbet vi kom, desto mere gik det op for mig, at jeg havde et helt anderledes perspektiv end kunden. Derfor har dette projekt givet mig forståelsen for vigtigheden af god kommunikation og hvor svært det kan være at sætte sig i kundens perspektiv. Udover dette har jeg selvfølgelig udvidet min

færdigheder indenfor primært web udvikling men også backend såsom rest services og databaser. Projektet har nemlig tvunget mig til at dykke ned i de forskellige muligheder, så man kunne finde de rigtige redskaber til vores projekt.

Kasper - Afslutningsvis er her min personlige konklusion for hovedopgaven. Set i bakspejlet er der en masse gode såvel som mindre gode elementer ved dette projektforløb. Med udgangspunkt i de negative aspekter først, vil jeg gerne komme lidt ind på struktureringen af arbejdet. Det har som det største projekt relativt til den givne tidsramme, været svært for os at se en ende på arbejdet, dette har ligget til grund i at jeg personligt, føler at vi har haft arbejdet i blinde til tider. Dette er sket som et resultat på manglende strukturering af arbejdet, vi har haft inkorporeret kanban boards, burndown charts og generelt bare haft et udadtil velfungerende informativ workspace. Men problematikken har været at vi har haft alle disse elementer, men ikke nødvendigvis brugt dem til deres fulde potentiale og ergo har vi ikke kunne uddeletere opgaver ordentligt for at kunne sikre at alle på et givent tidspunkt er sat til at arbejde på de højest prioriteret arbejdsopgaver.

For så modsat at vende blikket på de positive aspekter, skal Business Mindset have en tak. Det har været en rigtig god oplevelse at kunne sparre med en reel kunde. I tidligere studieprojekter, har vi ikke kunne tilegne os denne erfaring da vi har måtte lege vores egne kunder, hvilket ikke opfanger essensen af egentlig menneskelig kommunikation. Det kan i øjeblikket virke som en dårlig ting at vi som udviklere og vores kunde misforstår eller snakker forbi hinanden. Men virkeligheden er nu engang bare den at vi ved dette har fået os en mulighed for at forberede os selv på at tackle sværere udfordringer fremadrettet end hvis vi bare havde en udfordrings fri arbejdsproces.

KONKLUSION

[Forfatter: Kasper & Ian - læst/rettet af Christoffer og Jonas]

Hvordan kan man styre en udviklingsproces når man udvikler for en kunde bosat i et andet land

For at sikre en succesfuld udførelse af arbejdet har vi gennem vores projekt erfaret hvor væsentlig det er at være konkrete i vores kontakt til kunden. Så vi kan udnytte vores begrænset korrespondance med dem.

Dette kan vi konkludere da vi gentagne gange gennem forløbet har måtte dreje projektets retning, netop fordi vi ikke havde formået at forstå hele essensen af deres krav og fortolkede det forkert. Design sprints har været et godt værktøj til hurtigt visuelt at kunne vise kunden hvordan det er vi har fortolket deres efterspørgsler til produktet. Vi brugte design sprintet som en del af vores pre sprint og i sprint #1, her fik hurtigt værdi i form af om vi var på samme kurs.

Hvordan imødekommer man en kundes vision der ikke er konkret nok til at fastslå hvad deres krav er

Igen er kontakten til kunden det mest essentielle for at kunne finde frem til en konkret problemstilling for projektet. Hvis kunden kommer til os med en vision de har svært ved at kunne realisere, eller på anden vis problemer konkretisere. Er vi som fagfolk nødsaget til at sparre med kunden for at guide og vejlede dem til noget langt mere konkret som vi sammen kan arbejde os frem til i samarbejde.

Undervejs er vi dog på trods af vores vejledning af deres vision også nødt til at indse at vi er nødt til at kunne ændre slutmål, hvis det på et givent tidspunkt går op for kunden at dette alligevel ikke imødekommer deres forventninger til resultatet. Derfor er det også vores job at tage imod kravændringer og efterleve dem til bedste evne.

Hvordan kan man designe og udvikle et fundament, der har til formål at give Business Mindset en start platform der kan bruges til visualisering af succes metriker

Gennem vores to underspørgsmål kan vi altså konkludere at vi for at kunne designe og udvikle et produkt der imødekommer Business Mindsets behov, om et program der skal fungere som en start platform til deres store vision om slutproduktet.

Er vi nødsaget til kontinuert at holde kontakt til dem. Vi er nødt til at redegøre for de valg vi træffer og sikre en god sparring med dem for at sikre et produkt de fortsat kan benytte såvel som arbejde videre på fordi de ved hvordan det er designet og struktureret.

LITTERATURLISTE

Forfatter	Link	Benyttelse
Pre-sprint Kilder		
IKKE ANGIVET	tinyurl.com/yxbydrlx	Normalisering af database
IKKE ANGIVET	tinyurl.com/yy6kdbrr	Inspiration til test cases
“Midnight Spidy”	tinyurl.com/y4wa83fy	Research af design sprints
zFunx (YouTube)	tinyurl.com/y2fu4ul3	Research af CSS
Guru99 (YouTube)	tinyurl.com/y4oqb9yu	Research af test cases

Forfatter	Link	Benyttelse
Sprint #1 Kilder		
IKKE ANGIVET	tinyurl.com/oavrhp8	Hvordan CSS grid skal opsættes
Jesse Showalter (Youtube)	tinyurl.com/y6m3fp4	Mere om CSS grid
Will Soares	tinyurl.com/yxkxqpn	Fordele ved CSS grid

Forfatter	Link	Benyttelse
Sprint #2 Kilder		
IKKE ANGIVET (STACK-OVERFLOW)	tinyurl.com/yykrlktj	Implementering af webdriver wait
IKKE ANGIVET (GOOGLE)	tinyurl.com/yxq2omt2	Firebase dokumentation, brugt til at lave authentication/login
IKKE ANGIVET (GOOGLE)	tinyurl.com/y32qmglf	Firebase dokumentation, brugt til at styre brugere.
Miguel Vega (Youtube)	tinyurl.com/y3suljx4	Video vejledning til oprettelse af authentication.
The Net Ninja (Youtube)	tinyurl.com/yxnprxjb	Video vejledning til oprettelse af authentication.
IKKE ANGIVET (Codecademy)	tinyurl.com/y38l4pft	hvor er CORS
IKKE ANGIVET (Mozilla)	tinyurl.com/y7p7yv62	Implementering af CORS gennem http requests

Forfatter	Link	Benyttelse
Sprint #3 Kilder		
MBN Web docs	tinyurl.com/ntceq54	Forståelse for querySelectorAll
Stackoverflow	ti-nnyurl.com/y29ph695	removechild problemstilling
Stackoverflow	ti-nnyurl.com/y5mb93cb	Mere omkring removechild problemstilling
Stackoverflow	tinyurl.com/y4bzgvur	Target event. ja hvad er det? haha
Abeautifulsite	ti-nnyurl.com/y43h8hxo	Hvordan kan man slette dynamisk lavet elementer frem for statiske (som er li's lavet på forhånd i koden)
1bestcsharp.blogspot	tinyurl.com/y2pj6aay	Eksempel på delete i et listview
Youtube - 1 best csharp blog	ti-nnyurl.com/y5s8mfqy	Walkthrough delete methode, brugt til at få en bedre forståelse for hvordan det kan laves
isabel castillo - software dev.	ti-nnyurl.com/y653v6ac	Godt eksempel for stilfulde fejlbeskeder.
W3schools	tinyurl.com/yy8cwodf	Hvordan man Skjuler elementer på en side
Font Awesome	tinyurl.com/ya4b3zqv	Ikoner til fejlbeskeder

BILAG

OBS! Se alle eksterne bilag uploaded i wiseflow.

Bilag 1 - Design sprint #1 validate fase feedback

Hi Jonas,

Might be worth a call monday. I added some info to the plan did you get it?

Is a need for some additional interpretation.

Aim is to visualise the main elements of the team profile.

We should maybe start designing screens round the high level requirements I sent and take it from there.

There will need to be a team creation, company creation screen etc so maybe get that first and visualisation second.

Hope you have a good weekend!

Kindest regards,

Andrew

LOGBOG

Se externt i bilagsmappe