

[Link til Webløsning: https://cph-kundeservice.vercel.app/](https://cph-kundeservice.vercel.app/)

[Link til Github: https://github.com/Ballademager/cph](https://github.com/Ballademager/cph)

[Link til Prototype: https://www.figma.com/proto/](https://www.figma.com/proto/)



***Københavns Lufthavne***

**CPH**

**Hannibal & Mark**

Hannibal Grønne - hann800x@stud.kea.dk

Mark Hjort - mark204f@stud.kea.dk

Ca. 87.500 anslag

# Indhold

## Indledning

- Introduktion
- Problembeskrivelse
- Case
- Team Canvas
- Projektstyring
- Design Thinking

Side 1-7

## Empathize

- Design Research
- SWOT/TOWS
- Tænke Højt-test
- BERT
- Kvalitativt interview
- Lighthouse af nuværende site

Side 8-19

## Define

- Value Proposition Canvas
- Golden Circle
- Relevans Model
- Tone of Voice
- User Stories

Side 20-28

## Ideate

- Informationsarkitektur
- CPH Kundeservice Sitemap
- Card Sorting
- Idésortering
- Komponent idégenerering
- Sitemap vores løsning

Side 29-38

## Prototype

- Oversættelse af CVI
- Style Tile
- Wireframe
- Mockup
- Prototype
- Afgrænsning af løsning

Side 39-46

## Teknisk dokumentation

- Astro
- React
- CSS Styling
- NPM-pakker
- Eksempler fra koden
- Supabase
- Source Control
- Næste iteration

Side 47-62

## Test

- Lighthouse
- Page Weigth
- Wave Accessibility
- Tænke Højt-test
- Heuristisk Evaluering
- BERT

Side 63-71

## Refleksion

Side 72-73

## Litteratur

Side 74-75

## Bilag

Side 76-82



# → Indledning



# Introduktion

I denne rapport præsenterer vi vores arbejde med eksamensprojektet for 4. semester på MMD. Vi har i perioden udviklet en webløsning til en virksomhed, og har forudgående udført den relevante research for at danne rammen for opgaven. I denne rapport dokumenterer vi vores proces fra research, UI og designudvikling, test og indholdsproduktion til den færdige løsning samt refleksion over projektet.

Et af vilkårene for dette projekt var, at vi selv skulle finde en virkelighedstro case at arbejde med. Vi valgte at basere vores eksamenscase på et af gruppemedlemmernes praktiksteder, hvilket førte til, at vi valgte Københavns Lufthavn (CPH).

Københavns Lufthavn / CPH, er skandinaviens største lufthavn og fungerer som et vigtigt knudepunkt for mange internationale rejsende. Inden for CPH er afdelinger som Marketing og Customer Experience (MCX) en afgørende rolle i at sikre disse rejsende en god kunderejse og brandoplevelse, især når vi ved at der årligt passerer millioner af passagerer gennem CPH lufthavn. MCX-teamet er alle dedikeret til at forbedre de aspekter af kundeoplevelsen som de kan. Dette gælder alt fra digitalt engagement til fysisk navigation i lufthavnsområdet.

Afdelingens arbejde omfatter en bred vifte af opgaver, fra digitalt indhold, der engagerer brugere på tværs af platforme, til optimering af kunderejsen gennem lufthavnen med intuitive wayfinding-løsninger. Teamet bag dette inkluderer specialister i digitalt indhold, engagement, kundeoplevelse (CX), produktudvikling, digital strategi og brand management. Sammen udvikler de strategier, der ikke alene forbedrer den umiddelbare oplevelse for passagererne, men også styrker CPHs brand, som en førende aktør inden for luftfart.



# Problembeskrivelse

Marketing og Client Experience (MCX) varetager ansvaret for Copenhagen Airports (CPH) hjemmeside, som er en side der bliver besøgt af millioner af brugere årligt, og CPH's kundeserviceside er ingen undtagelse. MCX har under en evaluering af kundeservicesiden identificeret adskillige problemer, herunder forvirrende navigation, manglende essentiel information og fejlagtig "routing", som gør at brugeroplevelsen bliver forværret. Derudover får kundeserviceafdelingen mange henvendelser omkring områder, som de ikke kan hjælpe med.

Kundeserviceundersiderne er fordelt over en primær (cph.dk) og sekundær (cph-cem.my.site.com) webadresse, hvilket skaber forvirring og potentielt utryghed blandt brugerne. For at tackle disse problemer ønsker MCX at samle al kundeservicerelateret information på det primære domæne, udfase det sekundære og revidere indholdet, så det reflekterer de tjenester kundeservice faktisk kan tilbyde.

Det er værd at bemærke, at MCX er i gang med et større redesign af CPHs overordnede hjemmeside for at forbedre funktionalitet og æstetik. Dog er kundeserviceområdet ikke inkluderet i planerne.

Derved blev vores opgave at skabe en ny kundeservice side, for at skabe sammenhæng mellem CPHs nuværende side, og deres Corporate Visual Identity (CVI), samt revidere, optimere og bruge vores viden indenfor MMD til at forbedre UI / UX elementer på baggrund af relevant research.



## Case

Med dette projekt vil vi udarbejde en multimedie løsning for kundeserviceområdet, som er brugervenlig og intuitiv. Løsning vil foruden at integrere det nye redesign, forsøge at løse udfordringerne omkring informationsarkitektur og routing samt revidere indholdet, så det matcher fagområderne inden for kundeservice.



# Team Canvas

Vi arbejdede i et team af to medlemmer, hvor vi hver især bidrog med vores unikke styrker og kompetencer inden for multimediedesign. For at tydeliggøre disse kompetencer over for hinanden udarbejdede vi et Team Canvas for at sikre, at vi har en fælles retning og mål i vores gruppearbejde. Dette inkluderer både vores personlige mål og målene for projektet samt vores samarbejde.

Vi indskrev også vores individuelle styrker og svagheder i vores Team Canvas for at synliggøre hvad vi hver især har brug for i et samarbejde. For at få en bedre forståelse af hinandens arbejdsmetoder valgte vi at lave en Basadur test før projektets start. Testen viste, at vi havde en god blanding af styrker og evner.

For at optimere vores indsats evaluerede vi også på tidligere projekter og samarbejder inden opstarten. Dette hjalp os med at identificere områder hvor vi kunne forbedre vores arbejdsproces som vi som team og som individuelle medlemmer fik mest muligt ud af projektet.

## Team Canvas

Version 0.8 | [theteamcanvas.com](http://theteamcanvas.com) | [hello@theteamcanvas.com](mailto:hello@theteamcanvas.com)

Most important things to talk about in the team to make sure your work as a group is productive, happy and stress-free

Team name 
Date

<p><b>PEOPLE &amp; ROLES</b> </p> <p><b>Hannibal - Tovholder</b> [Implementer / Optimizer]</p> <p>Forholder sig kritisk til de valg vi træffer, og projektets status.</p> <p><b>Mark - Organizer</b> [Conceptualizer / Generator]</p> <p>Sikrer struktur på projektets status samt filer og opgaver.</p>	<p><b>COMMON GOALS</b> </p> <p>Vi vil anvende professionelle "source control" metoder, herunder branches, pull requests og code-reviews.</p>	<p><b>VALUES</b> </p> <p>Alle i teamet skal have en god forståelse af alle dele af processerne. Fleksible med hensyn til mødetider og god balance mellem studier og privatliv.</p>	<p><b>RULES &amp; ACTIVITIES</b> </p> <p><b>Skriftligt kommunikation</b> Messenger</p> <p><b>Mundtligt kommunikation</b> Microsoft Teams Fysiske møder [1+ ugentligt]</p> <p><b>Deling af filer</b> Github Figma Google Drive mappe</p> <p><b>Struktur redskab</b> Trello</p>
<div style="border: 1px solid black; border-radius: 50%; padding: 10px; display: inline-block;"> <p><b>PURPOSE</b></p> <p>Vi vil forbedre CPH Lufthavns kundeservice-side ved at reducere antallet af undersider og skabe konsistens på tværs af alle sider.</p> </div>		<p><b>PERSONAL GOALS</b> </p> <p><b>Hannibal</b> Have føling med og styr på projektet undervejs.</p> <p><b>Mark</b> Mere samarbejde gennem brug af branches og commits.</p>	<p><b>NEEDS &amp; EXPECTATIONS</b> </p> <p><b>Hannibal</b> Kræver struktur i projektet samt en vis overskuelighed.</p> <p><b>Mark</b> Kræver kommunikation og struktur gennem projektarbejdet.</p>
<p><b>STRENGTHS &amp; ASSETS</b> </p> <p><b>Hannibal</b> Stærk i kodning og arbejder effektivt og flittigt, når der er kodeopgaver på dagsordenen.</p> <p><b>Mark</b> Stærk i struktureret kodning, genererer ideer og har kendskab til flere grafiske værktøjer.</p>		<p><b>WEAKNESSES &amp; RISKS</b> </p> <p><b>Hannibal</b> Kan miste motivation, hvis opgaverne bliver ukonkrete eller uoverskuelige.</p> <p><b>Mark</b> Vil alt for meget og dvæler ofte ved det samme i lang tid, og er perfektionistisk.</p>	

Team Canvas by theteamcanvas.com. Created by Alexey Ivanov, Ekaterina Voroshilova  
 Team Canvas is inspired by Business Model Canvas by Strategyzer

This work is licensed under the Creative Commons Attribution-Share Alike 4.0  
 To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/>



# Projektstyring

Til at have overblik over vores opgaver brugte vi Trello, som er et visuelt projektstyringsværktøj, der bruger lister og kort til at hjælpe teams med at organisere projekter. Derudover lavede vi en 7 ugers plan i FigJam, hvor vi inddelte de forskellige uger i overordnede opgaver og delprocesser. Dette gav os muligheden for at holde opmærksomhed på projektets tidslinje, og på om vi var bagud eller foran vores estimering af projektets scope.



**Tilføj ikoner & billeder**  
in list Done

Members: HG, [Avatar] +  
Notifications: [Avatar] Watching

**Description** Edit  
+ 1 Officielt billede/ov | Download billeder

**Checklist** Hide checked items Delete  
100%  
[x] Opdater ikoner på Hittegods  
[x] Udskeft enten inder eller kontakt billede  
[x] Skal du bestille parkering? bilode  
[x] ikoner 2+2 index  
[x] ikoner 3+1 parkering  
[x] ikoner 3+1 security  
Add an item

**Activity** Show details  
[Avatar] Write a comment...

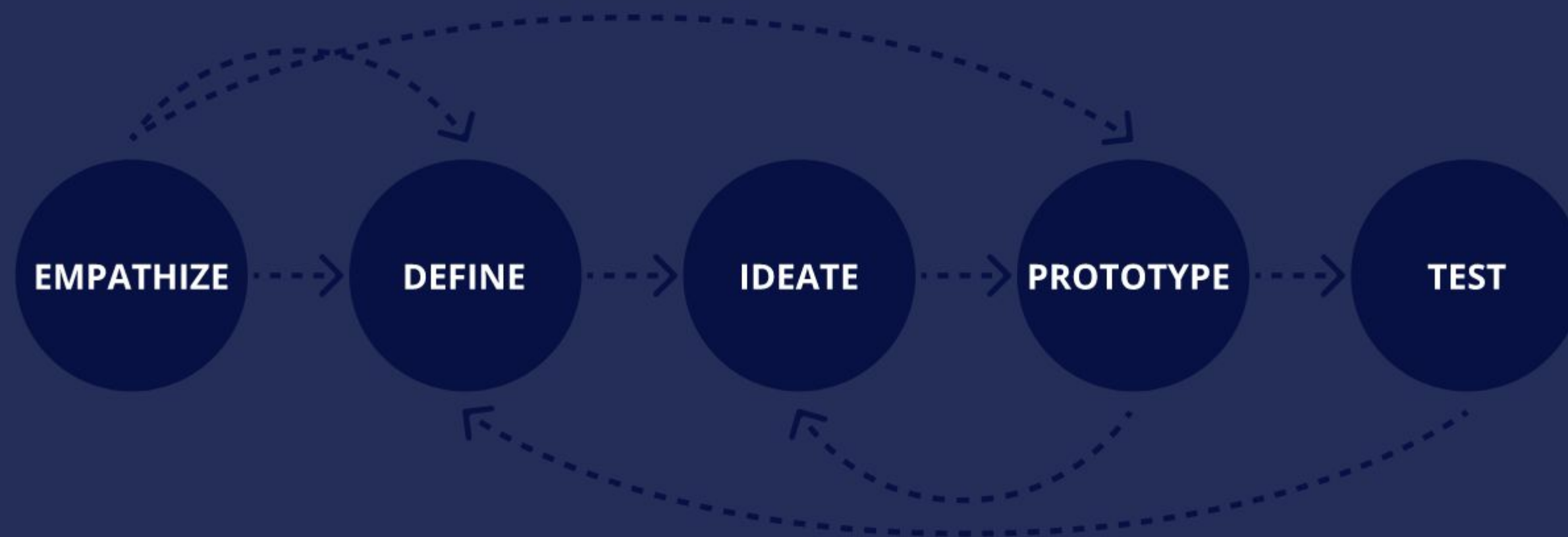
**Actions:** Move, Copy, Make template, Archive, Share



# Design Thinking

I dette projekt vil vi anvende Design Thinking<sup>1</sup> som udgangspunkt for vores designproces. Vi vil ligeledes anvende terminologien herfra til at inddele og placere de forskellige metoder og værktøjer, vi benytter undervejs, i de forskellige faser i modellen. Rapporten vil afspejle de faser og vi vil beskrive vores arbejde og proces med udgangspunkt i Design Thinkings faser.

Design Thinking er en løsningsorienteret procesmodel, som gennem fem faser, som kan understøtte udvikling af en gennemtænkt og gennemtestet løsning. Det er en ikke-lineær iterativ model, hvor designteamet ved behov har mulighed for at bevæge sig tilbage til tidligere faser, når ny indsigt er indhentet.





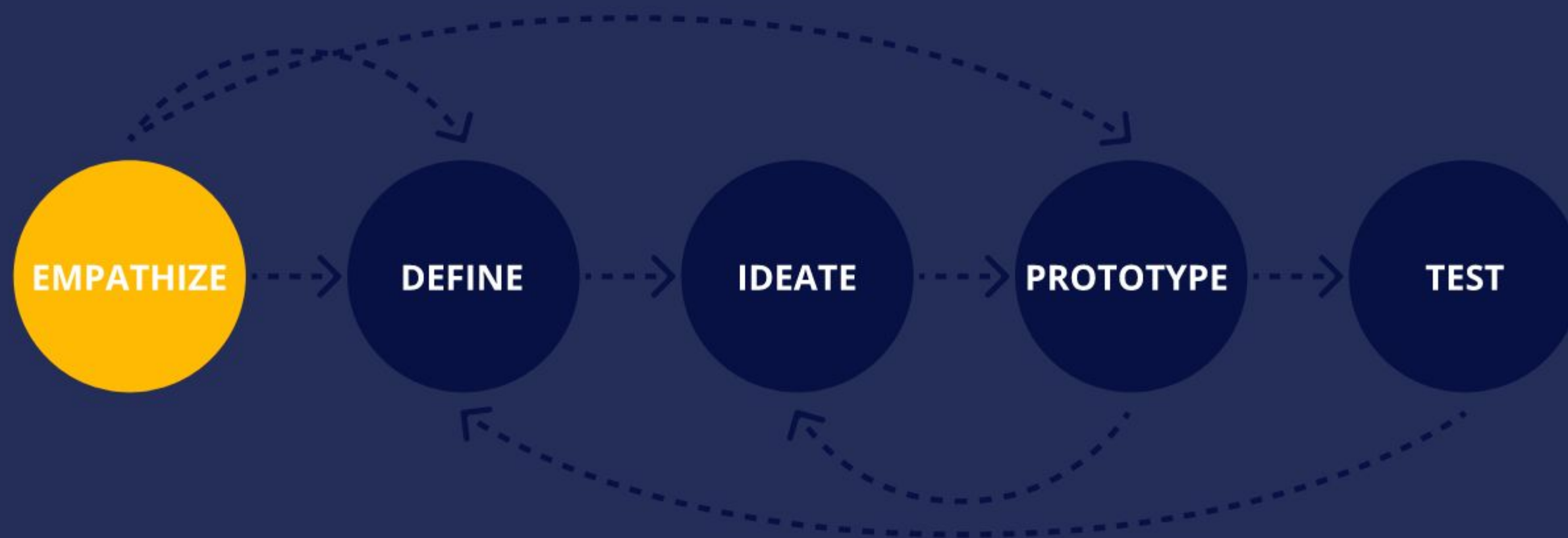
→ **Empathize**



# Empathize

Empathize-fasen er den indledende fase i Design Thinking-processen, hvor målet er at opnå en dyb forståelse for brugernes behov, perspektiver og oplevelser. Dette opnås gennem observation af brugerens interaktion med en løsning og gennemførelse af interviews. Ved at lytte og observere opnås indsigter, der danner grundlag for udviklingen af effektive løsninger.

Empathize-fasen er afgørende for at sikre, at løsninger faktisk er brugercentrerede og adresserer de egentlige behov og ønsker hos målgruppen.



Empathize

# Design Research

Design research er oftest noget den første research som foretages. Denne research bruges til at få en umiddelbar forståelse af det område man skal til at arbejde indenfor.

Den kan bruges til at få et indblik i, og eventuel inspiration fra, hvordan konkurrenter på markedet løser lignende udfordringer samt skabe en forståelse for trends og industristandarder.

Der findes flere metoder under design research og vi vil i det følgende præsentere dem vi har benyttet.

Københavns Lufthavne

CPH





## Product benchmark

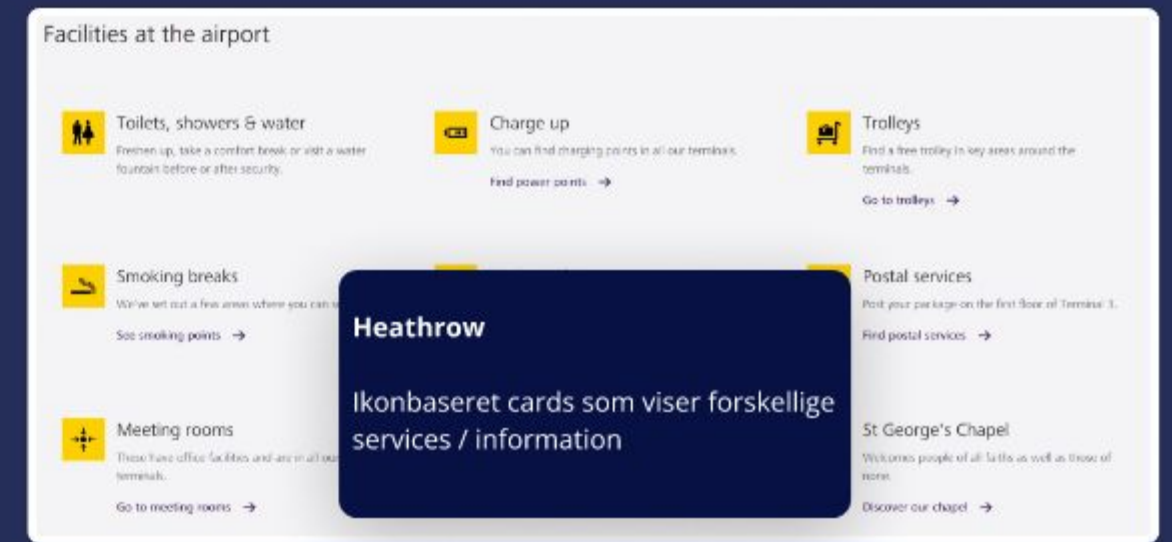
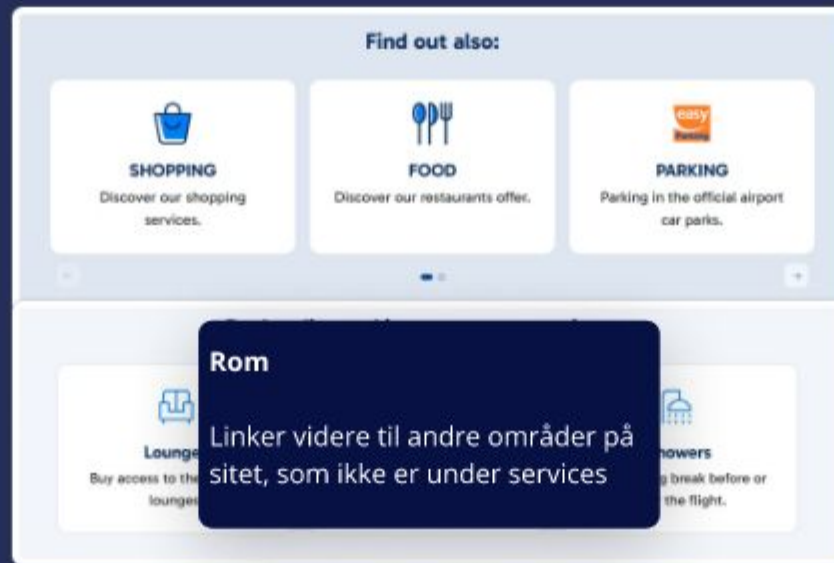
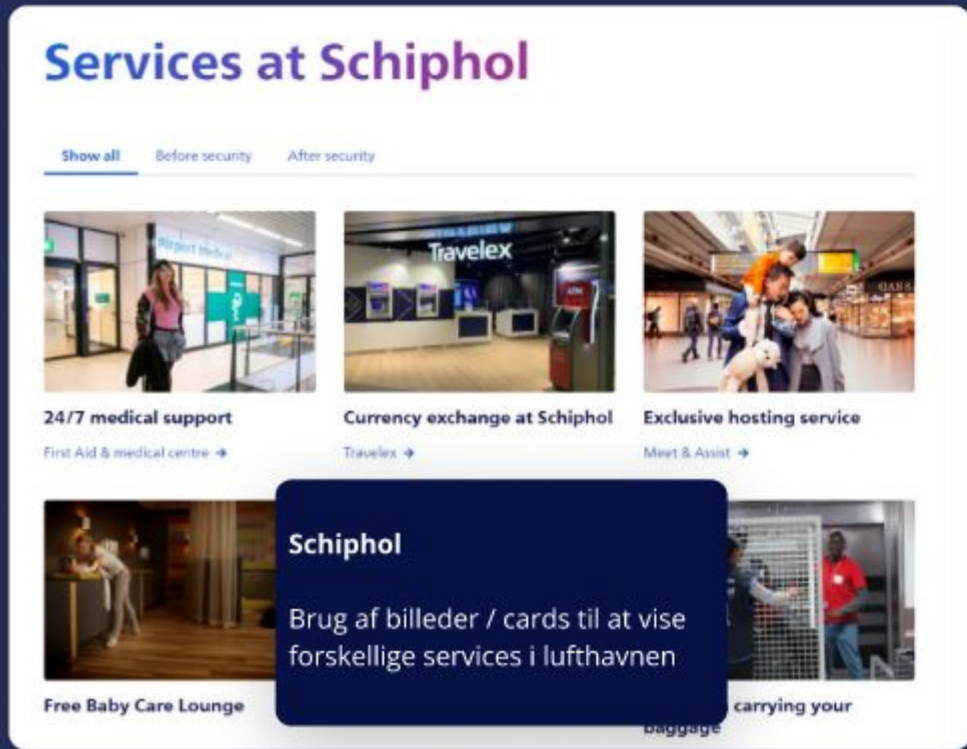
Product Benchmark<sup>2</sup> research bruges til at få en forståelse af det marked man designer til. I denne research udvælges vigtige konkurrenter for at få et overblik over deres designvalg, features og stil. Det giver en god forståelse for trends og standarder indenfor området.

Det er ligeledes en god metode til at få inspiration og starte den kreative proces.

Til vores Product Benchmark research udvalgte vi tre større lufthavne i Europa, som vi ved CPH Lufthavn måler sig mod. Vi så på deres løsninger ift. features og tog inspiration fra hvordan de løste nogle af de udfordringer vi vidste CPH Kundeservice havde.

Indsigter fra de forskellige lufthavne kunne samles til tre kerneindsigter:

- 1. Der bliver brugt mange billeder på siderne, typisk af mennesker.
- 2. Siderne har lys baggrund, med blålige sekundære farver.
- 3. Der bruges hyppigt cards til at vise muligheder eller links til undersider.





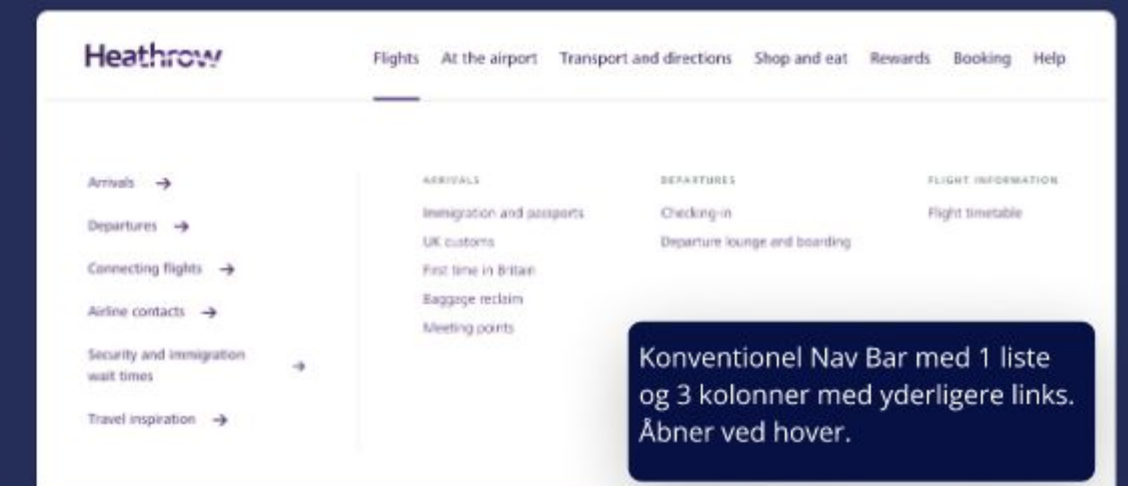
## Design Patterns

Design Pattern<sup>3</sup> research indebærer at undersøge og analysere etablerede designmønstre, som er velafprøvede løsninger på almindelige designproblemer. Denne research har til formål at undersøge eksisterende digitale løsninger for at få indsigt i hvilke interaktionsmønstre brugeren forventer og er vant til at navigere i. De genkendelige mønstre kan være med til at gøre brugeroplevelsen ubesværet og effektiv.

Vores research baseredes på de tre samme lufthavne, denne gang med et mere detaljeret fokus på sidernes UI-funktioner. Vi undersøgte Cards, Navigation og Itemlists.

Indsigter fra de forskellige lufthavne kunne samles til tre kerneindsigter:

1. I cards bruges enten billeder eller ikon/illustration sammen med overskrifter og tekst
2. To sider bruger dropdown-navigation. Dropdown'en åbnes forskelligt ift. hover/klik. En side bruger topmenu sammen med en burgermenu.
3. Der er tydelige ikoner bruges på hittegodsliste.





# SWOT / TOWS Analyse

En SWOT<sup>4</sup> analyse er et strategisk planlægningsværktøj, som bruges til at identificere de faktorer, som har betydning for en organisation eller projekts succes. Det er en struktureret metode til at forstå interne styrker og svagheder i organisationen samt de eksterne muligheder og trusler.

En TOWS<sup>5</sup> analyse anvender de identificerede styrker, svagheder, muligheder og trusler ved at kombinere de tilhørende udsagn.

## Strengths

- God til at vejlede over telefonen
- Gode anmeldelser for personalet i lufthavnen
- Brandkendelse, folk ved hvad de skal søge efter for at finde hjemmesiden (cph).
- Webside er både på dansk og engelsk og medarbejdere forstår de sprog

## Weaknesses

- Kunder kan ikke finde informationer på deres hjemmeside / Mange Undersider
- Dårlig Trustpilot score
- Det upraktisk at skulle ringe ind til kundeservice for at få svar
- Outdated content
- Potentiel langsom response tid
- Dårligt/intet samarbejde med andre områder i lufthavnen
- Mangel på personlighed / retter sig ikke målgrupperne (Business, Rejser, osv.)

## Opportunities

- Kan undgå unødige henvendelser, hvis kunder kan finde informationen online
- Ved at få folk til at dele deres gode oplevelser, kan man få lufthavnens ry til at være bedre
- Implementering af nye teknologier så som Machine Learning / AI
- Implementering af feedbackmuligheder, så folk føler sig hørt

## Threats

- Lufthavnen/kundeservice bliver forbundet med alle hændelser i lufthavnen og modtager kritikken.
- Folk som henvender sig har allerede en negativ forudindtagethed.
- Folk skriver ikke om deres gode oplevelser (TrustPilot), da det er forventning at det går problemfrit.
- Hjemmesiden skal overholde GDPR lovgivningen
- Kunder kan være fra hvor som helst i verden og derfor ikke tale dansk/engelsk som hjemmesiden vises på



Analysen førte til følgende vigtige indsigter:

- **Forbedring af informationsarkitektur:** Der er behov for at opdatere, forbedre og rydde op i informationen på undersiderne. Dette vil eliminere unødvendige oplysninger og gøre det nemmere for brugerne at finde de informationer, de har brug for.

- **Præcisering af oplysninger:** Det er nødvendigt at præcisere, hvilke oplysninger der kan findes hvor, og hvem der kan hjælpe, med de henvendelser der falder uden for kundeserviceområdet. Dette vil være med til at skabe en mere effektiv kundeservice.
- **Feedback og reviews:** Der er potentiale for at "prompte" brugere til at give gode anmeldelser ved positive oplevelser. Dette kan styrke CPH Lufthavns omdømme.

	Weaknesses	Strengths
Threats	<ul style="list-style-type: none"> <li>• (W1/T1): Ved brug af infobokse/FAQ, kan vi reducere dårlige oplevelser online og mindske kritik.</li> </ul> <div style="background-color: #1a2b3c; color: white; padding: 5px; margin: 5px 0;"> <ul style="list-style-type: none"> <li>• W5 / T2 - Ved lang response tid, kan den negative forudindtagelse kun blive værre.</li> <li>• W6 / T1 - Da at kundeservice ikke samarbejder mellem områder, men stadig bliver forbundet med alle hændelser af problemer bliver problemet kun større.</li> </ul> </div>	<ul style="list-style-type: none"> <li>• (S1/T1): Ved brug af infobokse/FAQ, kan vi undgå, dårlig kritik af lufthavnen, hvis kunden forstår at det ikke er inden for kundeservices fagområde.</li> <li>• (S1-S2/T2): Kundeservicemedarbejdere skal blive ved med at yde god service / vejledning over telefonen.</li> <li>• (S2/T3): Ved at lave et banner, som prompter kunden til at dele deres problemfrie oplevelser</li> </ul>
Opportunities	<ul style="list-style-type: none"> <li>• (O1/T3): Ved at lave en chatfunktion kan kunder hurtigt få svar på deres spørgsmål online.</li> <li>• (O3/T5): Ved at implementere AI / Machine Learning, kan man få alle sprog på siden.</li> <li>• (W1/O1): Ved at sortere i informationen kan kunden finde den rette information effektivt.</li> <li>• (W2/O4): Ved at implementere feedbackmuligheder på hjemmesiden, kan kunderne føler sig hørt og ikke have behov for at skrive negativt på TrustPilot</li> <li>• (W5/O1): Ved at lave informationen tilgængelig online, kan man holde response tiden nede.</li> </ul>	<ul style="list-style-type: none"> <li>• (S2/O2): Ved at lave et banner, som prompter kunden til at efterlade en god anmeldelse.</li> <li>• (S1/O1): Ved brug af infobokse/FAQ, kan vi fortælle, hvor kunden kan henvende sig, hvis det ikke er inden for kundeservices fagområde.</li> </ul>



# Tænke Højt-test

En Tænke Højt-test<sup>6</sup> er en metode til at teste brugervenlighed, hvor testpersonerne verbalt beskriver deres tanker og handlinger, mens de navigerer på en hjemmeside. Ved at dele deres umiddelbare reaktioner, overvejelser og udfordringer giver brugerne værdifulde indsigter som eventuelle udviklere og designere kan bruge til at vurdere hvordan deres webside opleves.

I forbindelse med vores undersøgelse og test af den nuværende hjemmeside, valgte vi at lave en serie af Tænke Højt-test med 3 forskellige testpersoner. Dette havde formålet at indsamle deltagernes oplevelser ved interaktion med CPHs nuværende hjemmeside.

Inden testen udarbejdede vi en række spørgsmål og opgaver, som testpersonerne skulle gennemføre under navigation af hjemmesiden, mens de "tænke højt". Testen blev optaget og der blev taget noter undervejs.

Efterfølgende organiserede vi alle resultaterne i vigtige indsigter, som vi senere kunne bruge som til at optimere vores egen løsning.

Med de udvalgte spørgsmål kunne vi undersøge informationens tilgængelighed og sidens navigerbarhed.

**"Du har booket parkering i Københavns Lufthavn men skal ikke længere bruge din booking. Kan du finde ud af hvor du aflyser din parkering?"**

**"Du har mistet din telefon og vil gerne have den retur."**

**"Du vil vide hvad Tax-Frees åbningstider er, kan du finde denne information?"**

Vores resultater viste at alle deltagerne var kritiske når det kom til mængden af undersider der blev brugt, og hvordan det fik kundeservice siden til at virke rodet. Disse indsigter var yderst vigtige når det kom til udviklingen af vores egen løsning, da de kunne informere os om og hjælpe os med at undgå nogle af de udfordringer det eksisterende site har, som vi bør prøve at undgå.

**Informationen på CPHs Kundeservice side er meget spredt, og man skal flere sider dybt for at finde den information man reelt søger**

**Informationen opleves besværlig og tidskrævende at finde**

**Hittegodstabelen har meget begrænset information om de enkelte genstande**



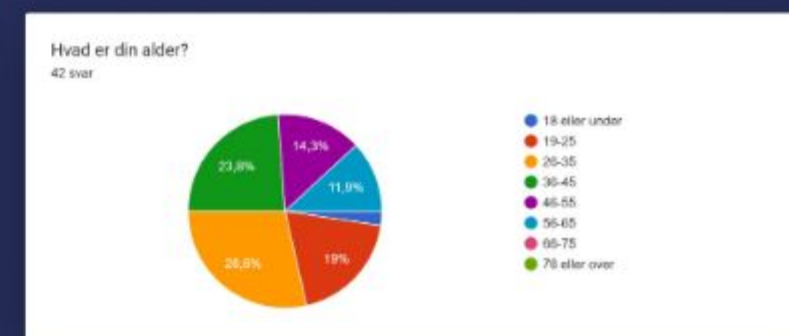
# BERT

Vi udførte en BERT<sup>7</sup> (Bipolar Emotional Response Test) som vi brugte til at vurdere den eksisterende kundeservice side. Formålet med denne test er at undersøge en brugergruppes følelsesmæssige reaktioner og opfattelse af et produkt eller en side. I testen opstilles nogle modsatrettede adjektiver eller udsagn, som testpersonerne skal bruge til at beskrive deres opfattelse af det som testes.

Vi oprettede en Google Form for at gennemføre testen med så stor en gruppe testpersoner som muligt. Vi udvalgte seks modsatrettede udsagn, som vi ønskede at undersøge brugernes opfattelse af CPHs eksisterende sites på baggrund af.

Vi spurgte også MCXs Digital Strategy Manager, Rebecca, hvor hun ønskede at kundeserviceside skulle ligge mellem disse modpoler, for at give os en målsætning at stræbe efter for vores egen løsning.

På billedet kan vores resultater fra vores BERT ses. Vores test fik 42 besvarelser, som fordelte sig bredt på baggrund af alder.



Uoverskuelig

Overskuelig



Resterende af vores BERT resultater kan ses i Bilag 1



# Kvalitative Interview

I den opstartende fase af projektet udførte vi to kvalitative interviews med interessenter fra CPH.

Vi interviewede Rebecca, som er Digital Strategy Manager i MCX og er ansvarlig for CPHs digitale strategier samt tilsyn med hjemmesiden. Interviewet med Rebecca skulle være med til at informere os om den større kontekst, som kundeservice er en del af og lære om ønsker fra organisationens ledelse samt hvilke udfordringer der på forhånd var identificeret fra afsenderens side.

Rebecca så det som et af de mest gennemgribende udfordringer, at man som bruger bliver ført frem og tilbage mellem den primære webadresse [cph.dk/kundeservice](https://cph.dk/kundeservice) og den sekundære webadresse [cph-cem.my.site.com](https://cph-cem.my.site.com). Det sekundære domæne indeholder et stort FAQ-system og er også der hvor funktionaliteter som kontaktformularen befinder sig. Sitet vedligeholdes externt og adskiller sig i høj grad fra det primære site hvad angår visuel- og skriftlig kommunikationsstil. Rebecca udtrykte et klart ønske om, at alt samles på hovedsiden, for at skabe en mere sammenhængende brugerrejse.

## Interviewet med Rebecca gav os følgende indsigter:

- Det er problematisk at man som kunde bliver sendt rundt til flere sites, som er forskellige ift. visuel stil, information, navigation.
- Der er forskellig sprogbrug/ToV på de to kundeservicesider.
- 0.7% af [cph.dk](https://cph.dk)'s brugere besøger [cph.dk/kundeservice](https://cph.dk/kundeservice).
- Den samlede vision og retning af alle ønsker er udfordrende, da mange afdelinger er inde over hjemmesiden.
- De forskellige dele af sitet, som henvender sig til forskellige målgrupper som corporate (journalister, jobansøgere), business(butikker, partnere) og kunder (rejsende), kan godt have forskellige versioner af CVI'et.



Derudover foretog vi et interview med kundeserviceagent Emma, for at lære om nogle af de lavpraktiske udfordringer medarbejderne oplevede og kendte til for hjemmesiden. Emma kunne berette om flere problemer det nuværende site havde, som hun og hendes kollegaer mærkede i deres arbejde. De oplever, at rigtig mange ringer ind fordi de ikke kan finde det de søger på siden. Mange ringer med forespørgsler, som ikke falder inde for kundeservices ekspertise og ender med en viderestilling.

Derudover har brugerne svært ved at finde, forstå og anvende hittegodslisten samt at skelne mellem hvor de forskellige typer genstande ender.

Vi foretog også et besøg hos hittegodsaftningen, som foregik som et uformelt interview, hvor vi spurgte ind til deres vilkår og hvordan de interagerer med databasen bag hittegodslisten.

De havde ligesom Emma oplevelsen af at brugerne har svært ved at bruge listen korrekt. Brugeren ender ofte med at vælge en mere eller mindre tilfældig genstand på listen og derefter beskrive deres genstand.

### Interviewet med Emma gav os følgende indsigter:

- Kunder kan ikke finde det de leder efter inde på kundeservicesiden og det er nemmere for dem bare at ringe.
- Størstedelen af opkaldene er "spild af tid".
- Kundeservice vil helst besvare henvendelser via email/formular.
- Kunder finder hittegodssiden, men finder ikke listen.
- Kunder har svært ved at bruge listen.
- Der kan muligvis tilføjes dato / timer for registrering for hver genstand.
- Kunder forstår ikke, hvad der bliver i lufthavnen og hvad der bliver sendt videre.

### Indsigter fra Hittegods:

- Det fremgår ikke af hjemmesiden, at de donerer genstande til Blå Kors.
- Hittegodslisten mangler visuelt understøttende elementer.
- Det er svært for brugeren at gennemskue, hvor de forskellige typer genstande ender.

Disse interviews og samarbejdet med afsenderen har været essentielt for at informere os om hvilke problematikker den nuværende hjemmeside har. Disse problematikker er vigtige at have for øje, når vi skal idégenerere og bygge en prototype.



# Lighthouse af nuværende site

Vi valgte at lave en Google Lighthouse-test<sup>8</sup> på CPHs nuværende kundeserviceside, for at identificere områder og optimeringsmuligheder. Formålet med denne test er at vi kan skabe en forståelse for de udfordringer som den nuværende kundeserviceside står over for, og få identificeret de områder vi kan forbedre i vores digitale webløsning.

Som det kan ses på graferne, er SEO og Accessibility scoren lav i forhold til at det er en kundeservice side for et stort firma så som CPH Lufthavne. Det bemærkes også at Performance er meget lav.

63

## Performance

- Reduce unused Javascript
- Reduce unused CSS
- Defer offscreen images
  - Billedet der vises hvis videoen ikke afspilles bliver loadet først istedet for lazy load.
- Serve images in next-gen formats
  - Webservice bruger png / jpg

73

## Accessibility

- Buttons do not have accessible name
- Image elements do not have [alt] attribute
- Link do not have a discernible name

93

## Best Practices

- Browser errors were logged to the console
  - "Failed to load resource"
- Issues were logged in the Issues panel in Chrome Devtools

75

## SEO

- Document does not have a meta description
  - Description text is empty
- Image elements do not have [alt] attributes
- Links are not crawlable



→ Define

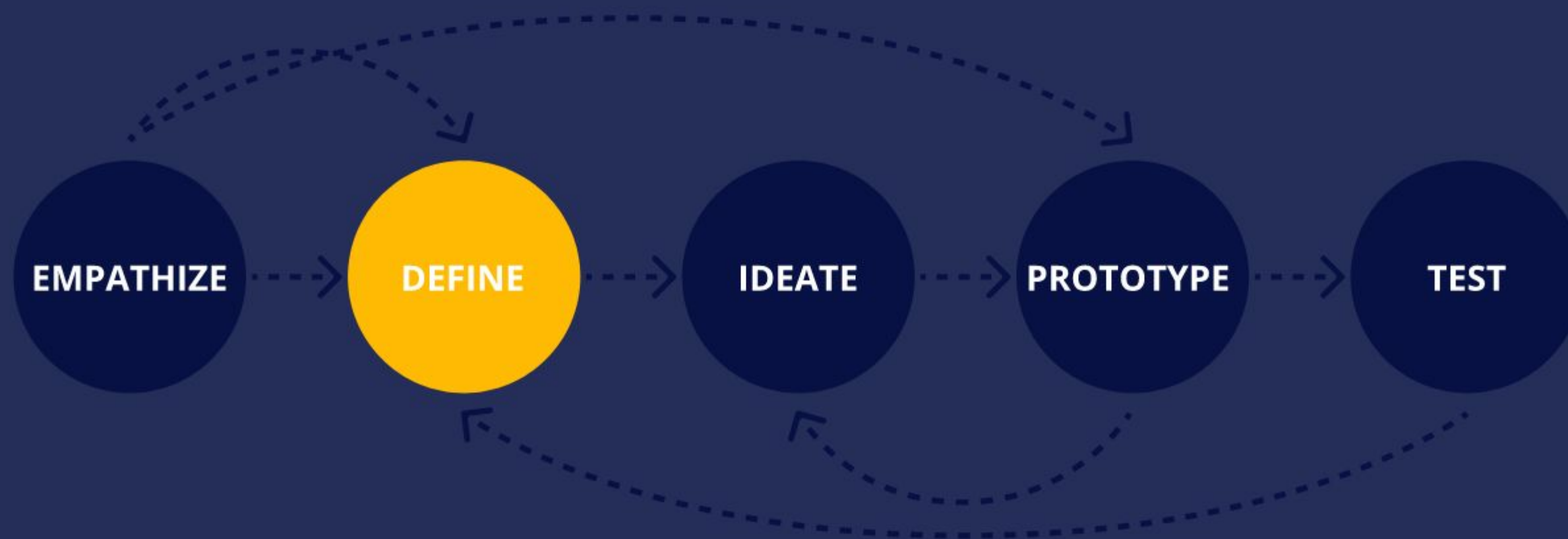


# Define

I Define-fasen organiseres den information og de indsigter, som er indhentet i Empathize-fasen.

Indsigterne fra Empathize-fasen organiseres og kategoriseres for at identificere mønstre og fælles temaer. Herefter sammenholdes lignende observationer og indsigter, så en klar forståelse af de primære udfordringer bliver tydelig.

Denne fase er afgørende for at skabe klarhed og retning for det videre arbejde i designprocessen.





# Value Proposition Canvas

Value Proposition Canvas<sup>9</sup> er en model, der fokuserer på kundesegmentet, og den værdi virksomheden kan skabe for dem. Den kan hjælpe med at sikre, at virksomhedens værdier passer til de behov deres kunder har.

Modellen er delt op i to dele: højre del, kaldet "Observe", indeholder kundesegmentet og de indsigter vi har observeret hos dem, såsom deres problemer (pains) og det, de vil opnå (gains). Det er også her vores målgruppe begynder at tage form, men da vi arbejder med en kundeserviceside, har vi ikke en specifik målgruppe at sigte efter.

Den venstre del af modellen, kaldet "Design", indeholder de værdier som virksomheden kan give kunderne, både i form af produkter og hvordan de kan hjælpe kunder med deres "pains" og "gains".

Modellen viser os, at CPH Lufthavn har rig mulighed for at afhjælpe nogle af de problemer (pains), som kunderne oplever ved at have en webløsning, der kan hjælpe de rejsende med at finde den nødvendige information hurtigt online.

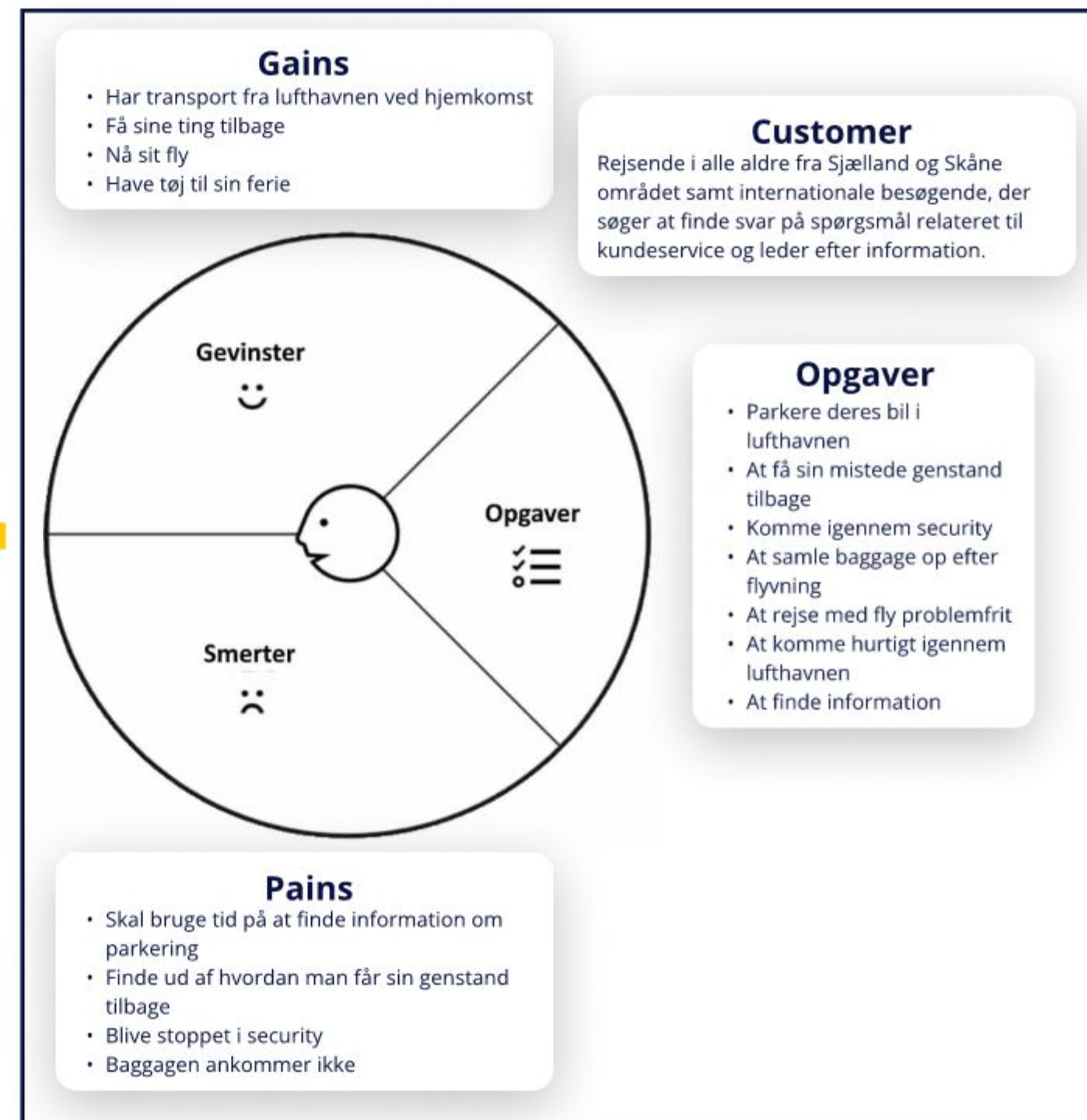
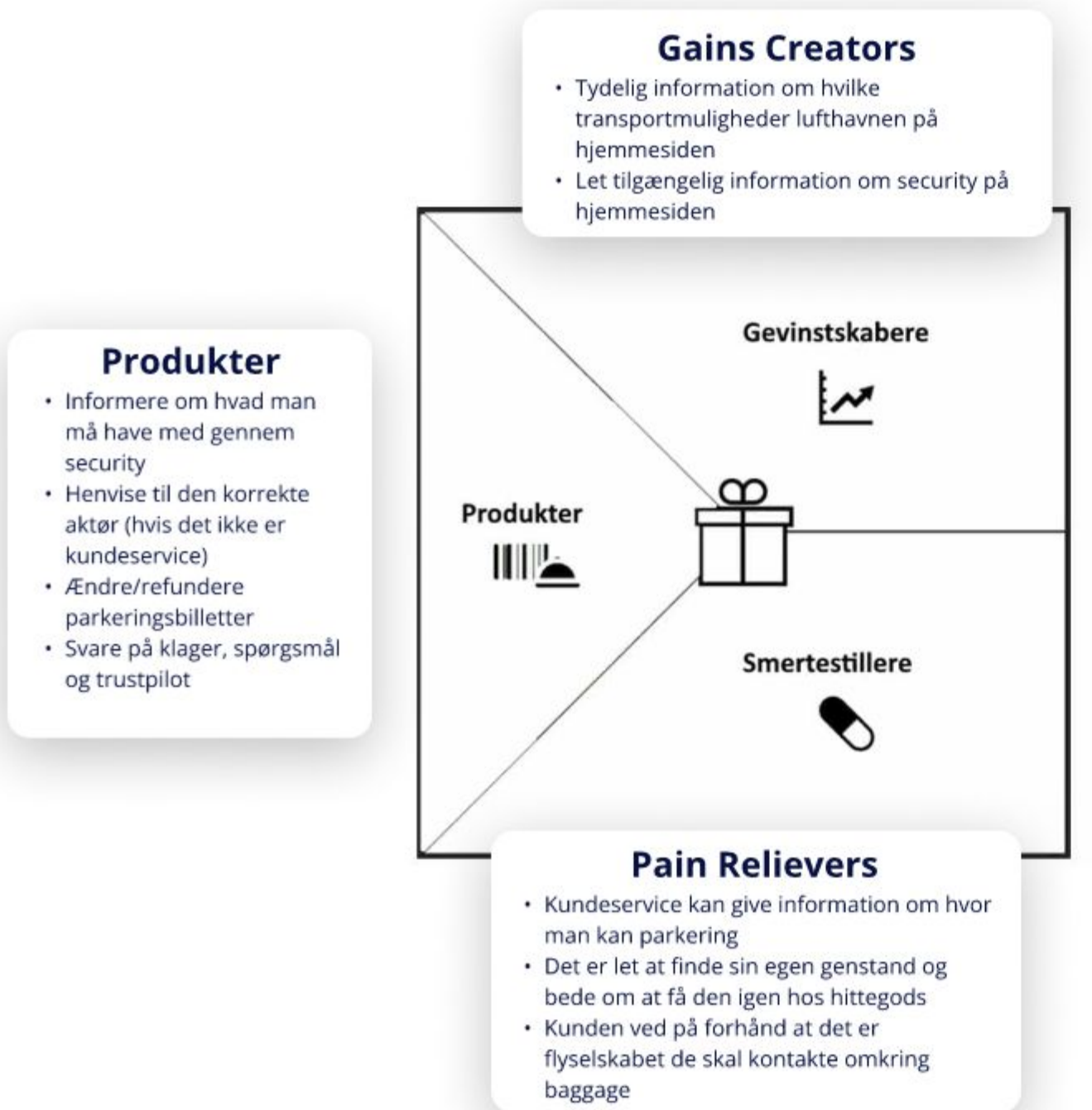
Den udfyldte model kan ses på næste side.





## Design

## Observe



## Value Proposition Ad-lib

“Vores multimedieløsning for kundeservicesiderne på cph.dk hjælper lufthavnens rejsende, som har brug for information eller ydelser i lufthavnen til hurtigt at finde den fornødne information på hjemmesiden.”



# Golden Circles

For at finde ud af vores formål og hvordan vi skal tilgå CPH Lufthavns kunder, valgte vi at bruge Astrid Haugs Content Strategy<sup>10</sup>, hvor første step omhandler formål. Her bruges Simon Sineks The Golden Circle<sup>11</sup>, som ved at spørge Hvorfor, Hvordan og Hvad, kan hjælpe en virksomhed med at udpege dens formål.

Selvom vi har valgt ikke at lave socialt medie indhold til CPH Lufthavns hjemmeside eller brand, mente vi bestemt at de første steps i Astrid Haugs Content Strategy vil give os en masse information om hvordan vi bedst muligt kan formidle oplysninger på en kundeserviceside.

## Why

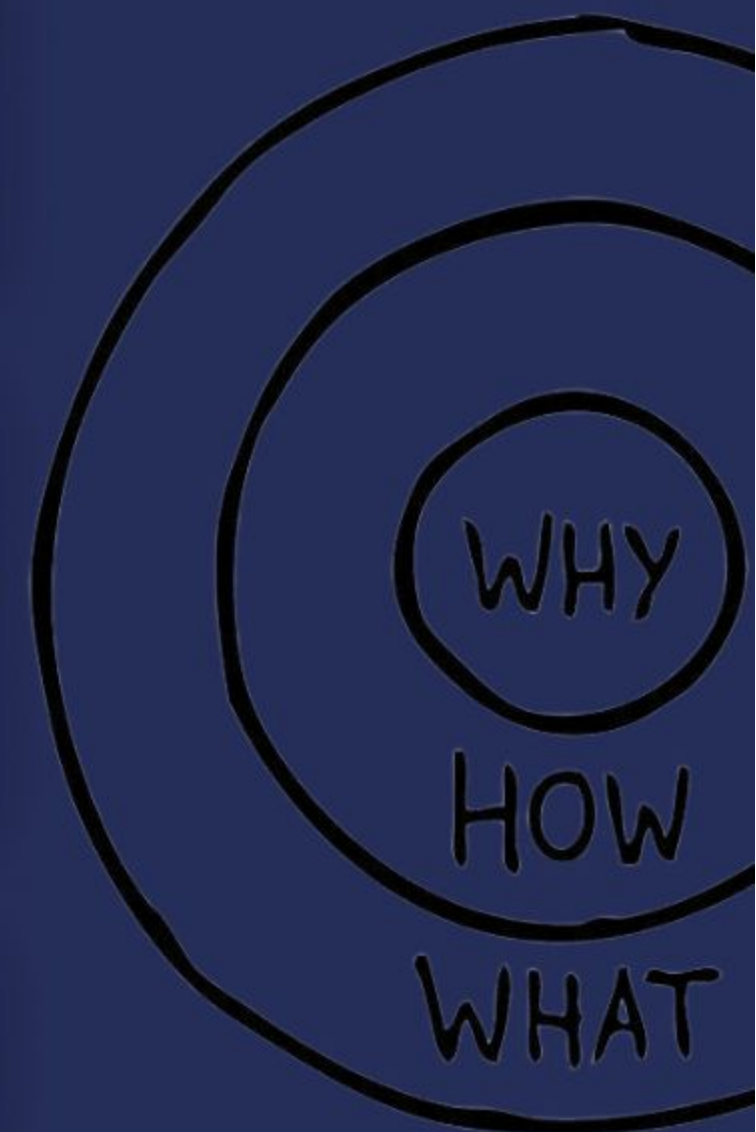
København Lufthavns Kundeservice skal være med til at skabe en god rejse oplevelse, samt gøre den så problemfri og behagelig som muligt.

## How

Dette gøres ved at kommunikere tydeligt og empatisk både telefonisk og skriftligt, så passagerer føler sig trygge og velinformeret.

## What

København Lufthavns Kundeservice tilbyder kunder hjælp med parkering, hittegods, security-spørgsmål og generel information, både telefonisk og online.



10 - Astrid Haugs Content Strategy

11 - Simon Sineks Golden Circle



# Relevans Model

Næste trin i Astrid Haugs Content Strategy går ud på at skabe relevans for kunderne på CPH Lufthavns kundeserviceside. Dette gøres ved at se på indsigter fra vores kundesegment og deres behov og sammenligne dem med indsigterne fra CPH Lufthavn og deres ønsker.

Det er afgørende at identificere emner og information, der ikke kun er relevant og værdifuld for kunderne, der anvender kundeservicesiden, men som også passer ind med CPH Lufthavns mål.

Dette sikrer, at det indhold, der præsenteres på vores webløsning, ikke blot imødekommer kundernes behov, men også bidrager til lufthavnens overordnede succes ved at være balanceret mellem brugbarhed for kunderne og målrettet for CPH Lufthavn.

Vi kom frem til flere relevante punkter og ideer, som vi satte sammen til nogle handlingsorienterede forslag, som kan ses på modellen nedenfor. Med disse indsigter og handlingsforslag kunne vi danne et godt fundament til en brainstorm til vores egen webløsning i forhold til komponenter og hvilke funktionaliteter de skal indeholde.





# Tone of Voice

For at informere os om sidernes copy, foretog vi en Tone of Voice-analyse<sup>12</sup>. Analyse er en undersøgelse af, hvordan en organisation eller et brand kommunikerer med sine målgrupper gennem skriftlig og mundtlig kommunikation. Analyse fokuserer på de sproglige elementer, der anvendes i kommunikationen, såsom ordvalg, formelle eller uformelle vendinger, og den overordnede følelsesmæssige tone. Til at foretage analysen udvælges nogle nøgleord, som beskriver organisationens ønskede udtryk. Disse placeres i et skema, hvor de beskrives og der tilknyttes eksempler på "DO's" og "DON'Ts" til hvert ord.

For at finde nøgleordene, som skulle inspirere sidernes copy-stil, måtte vi overveje afdelingens rolle for brugerne.

Vi var opmærksomme på at afsenderen er en kundeserviceafdeling, derfor måtte sproget være venligt. Derudover overvejede vi også, hvordan tone of voice kan påvirke brugeroplevelsen. Et konsekvent og veldefineret sprog kan øge brugernes tillid og tilfredshed, hvilket er særligt vigtigt for en kundeserviceafdeling.

Fra interview med medarbejdere vidste vi at mange ringede ind til kundeservice fordi de ikke fandt den nødvendige information. Derfor var det vigtigt at holde sproget informativt og tydeligt samt kortfattet, så brugeren nemt og hurtigt kan scanne teksten og vurdere om de læser det rigtige sted.

12 - Tone of voice

Karakteristika	Beskrivelse	Do	Don't
<b>Informativt</b>	Vores tekst er informativ og designet til at hjælpe brugeren hurtigt videre.	Vær deskriptiv og gå lige til sagen. Giv brugeren et link/telefonnr. hvis der henvises til en anden aktør.	Bruge pyntende adjektiver.
<b>Tydeligt</b>	Vores tekst er tydelig og præcis, uden unødigt fyld, så brugeren nemt kan overskue den og hurtigt handle på informationen.	Brug ikoner til at forstærke tydeligheden af handlingers (tryk på knap/link) konsekvenser. Skriv hvad der findes på en underside.	Brug af humor, sarkasme og andet der kan misforstås. Brug fylde tekst uden formål. Brug af links uden der står hvor det fører hen.
<b>Venligt</b>	Vi er serviceminded og venlige for at give vores brugere en god oplevelse.	Neutralt, men venligt. Brug enkelt sprog.	Skrive kommanderende.
<b>Kortfattet</b>	Brugeren har muligvis travlt og andet at tænke på og skal ikke bruge unødigt tid på at scanne fyldig tekst.	Brug nøgleord til at lede brugeren til det de søger: "Reservér din plads her" og "Se <b>Parkeringsmuligheder</b> her"	Skrive tekst uden formål. Brug af avanceret sprog/fremmedord.



# User Stories

Ved brug af informationen fra tidligere analysemodeller kan vi nu skrive User Stories<sup>13</sup> ved brug af den information. Fordelene ved at bruge User Stories som et analyseværktøj er, at vi skaber et mere brugercentreret fokus, hvor vi prioriterer de reelle brugerbehov og fokuserer på at opfylde dem.

Formålet er at formulere, hvorledes en indholdsfunktion skaber værdi for brugeren. Dette hjælper os til at forstå og imødekomme CPHs brugere og deres behov. Anvendelsen af User Stories kan være med til at gøre udviklingen mere brugercentreret, da der fokuseres på brugerens behov i interaktion med den teknologi, teamet er ved at udvikle, hvilket kan understøtte en forbedret brugeroplevelse.

I vores "User Stories" har vi fokuseret på de rejsende og kundeserviceagenterne, hvilket giver os et indblik i hvordan vi kan hjælpe begge sider af interaktionen.

Til højre, ses vores "User Stories", hvor vi har markeret dem, vi mente var de vigtigste for vores case."

Som rejsende vil jeg vide, hvem jeg skal kontakte for at finde min mistede bagage, så jeg kan komme hjem med alle mine ting i behold.

**Som kundeserviceagent vil jeg give rejsende den information som de ønsker for at yde en god service.**

Som en passager ønsker jeg nemt at kunne finde information om sikkerhedsprocedurer, så jeg kan passere igennem sikkerhedskontrollen problemfrit.

Som en rejsende ønsker jeg nemt at kunne finde information om lufthavnens faciliteter, så jeg kan navigere effektivt rundt på lufthavnen.

Som rejsende vil jeg have lettilgængelig information, så jeg kan undgå tvivl.

Som kundeserviceagent vil jeg dirigere rejsende til de rigtige kontaktpersoner hvis jeg ikke selv kan svare på eventuelle spørgsmål.

**Som rejsende, vil jeg have min glemte computer tilbage, så jeg ikke skal købe en ny.**

Som rejsende, vil jeg finde kundeservices telefonnummer, så jeg kan spørge om hjælp.

Som rejsende, vil jeg forlænge min parkering, så jeg undgår en parkeringsbøde.

**Som en rejsende, vil jeg have information om security og hvad jeg må have i min bagage, for hurtigt at kunne omgås i lufthavnen.**

Som en rejsende ønsker jeg at kende til spisemuligheder i lufthavnen, så jeg kan få et måltid under min mellemlanding.

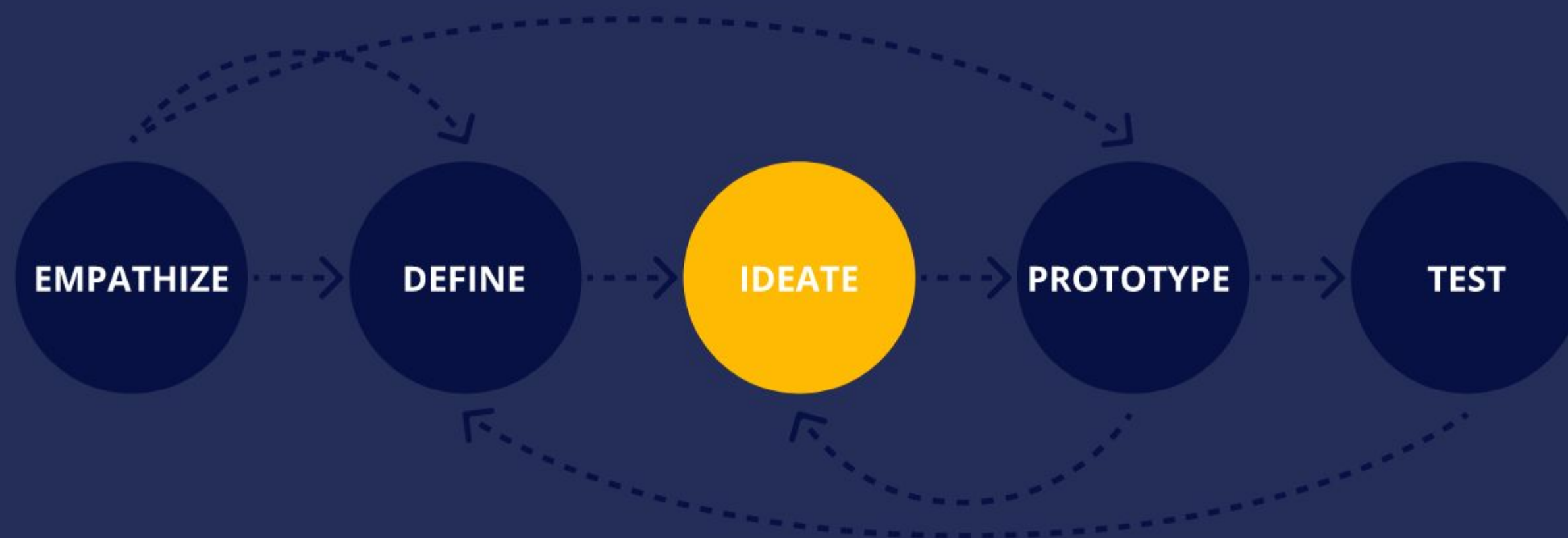




# Ideate

Ideate-fasen handler om at generere idéer til løsninger på de identificerede udfordringer. Ved at bruge forskellige idégenereringsværktøjer og -teknikker opfordres teamet til at tænke bredt og kreativt.

Det er afgørende at have så mange idéer som muligt for at sikre, for at have en bred vifte af muligheder, der kan bruges til at løse de identificerede udfordringer i udviklingen af prototyper.





# Informationsarkitektur

Før vi begyndte ideationsprocessen, undersøgte vi informationsarkitekturen for CPH's kundeserviceside. Formålet var at klarlægge indholdet og synliggøre eventuelle problemer ved den originale hjemmeside samt at strukturere vores egen løsning med fokus på brugerne. Vi ønskede at skabe et logisk og intuitivt informationsmiljø, der gør det nemt for brugerne af CPH's kundeserviceside at få adgang til den information de har brug for.

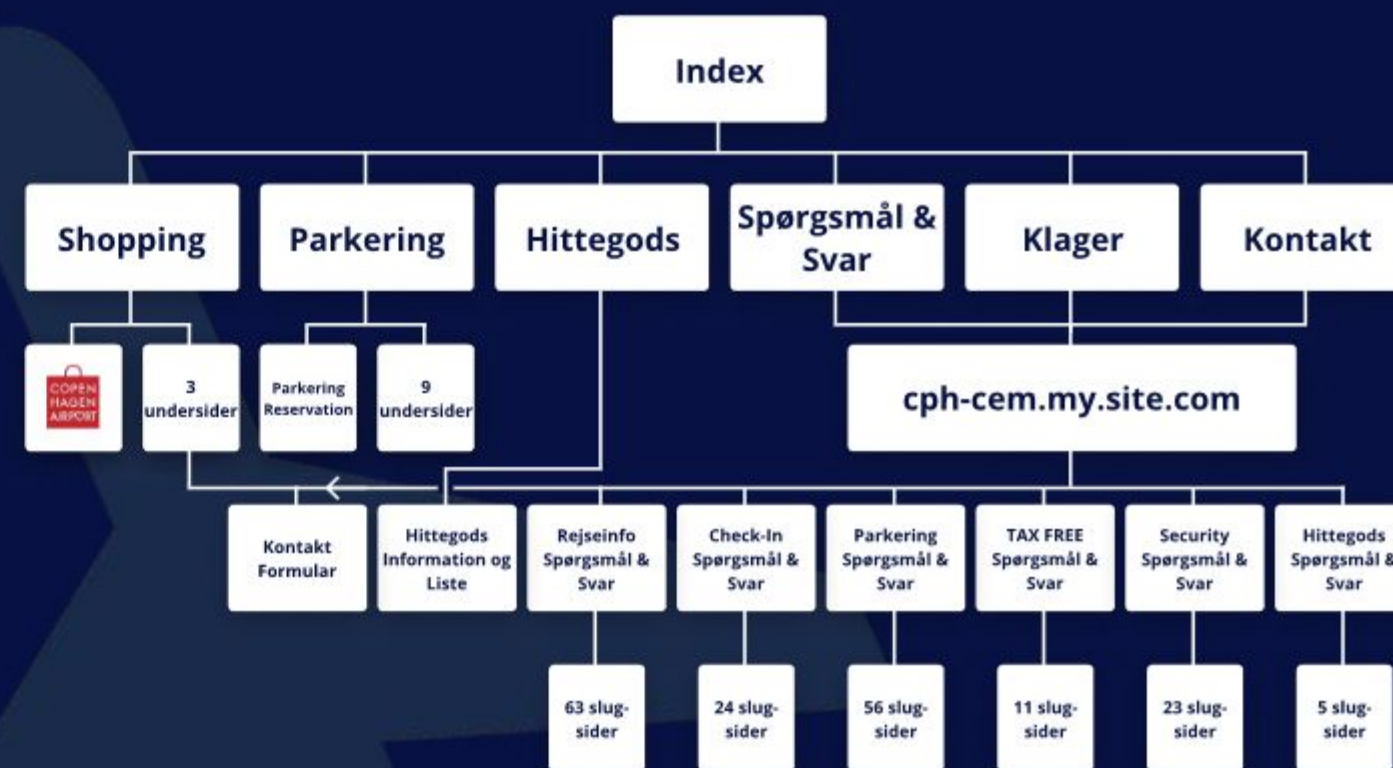
Vi startede ud med at lave et Sitemap af den originale side for at skabe os et overblik over sidens struktur.

## CPH Kundeservice Sitemap

Københavns Lufthavns kundeserviceside har seks links på forsiden, hvilket giver indtryk af, at der er seks undersider, dog bruges halvdelen af de seks valgmuligheder til at sende brugeren til en sekundær side, der ikke er en del af cph.dk.

Den sekundære side, også kaldet "Salesforce", fungerer som et forum/FAQ, der besvarer ofte stillede spørgsmål så nøjagtigt som muligt. Dog er der op mod 150 undersider, der hver især besvarer forskellige spørgsmål, hvilket kan gøre det svært for brugerne at finde den konkrete information, de leder efter.

### CPHs Kundeserviceside





# Card sorting

Med informationen fra CPH's kundeservicesides sitemap kender vi nu til, hvordan websiden er struktureret. Derfor valgte vi at bruge Card Sorting til at sortere og organisere den information vi vil have på vores egen webløsning.

Card Sorting er en brugervenlighedstestmetode, der ofte bliver brugt i designprocessen af websites og andre digitale produkter. Vi valgte at lave en hybrid sortering, da vi havde forudbestemt nogle af vores kategorier, på baggrund af den eksisterende informationsarkitektur, men vi samtidigt ville forholde os åbent til eventuelt nye grupperinger.

De forudbestemte kategorier vi havde var; Forside, Hittegods og Kontakt / Spørgsmål & Svar. Det blev dog hurtigt klart, at vi manglede en gruppering til alt information vedrørende parkering, og efter lidt intern diskussion valgte vi også at oprette en kategori ved navn Security, da vi tidligere i vores interview med Emma var blevet fortalt at en stor mængde spørgsmål vedrører dette område.

Billedet nedenfor viser vores færdige Card Sorting, hvor det kan ses at vi har tilføjet en FAQ, en omdirigering liste og andre komponenter. Disse komponenter kan vi nu idégenerere på og komme med forslag til, hvordan de skal se ud og fungere på vores webløsning.





# Idésortering

Gennem ideations-fasen og de forskellige anvendte værktøjer opstod mange forskellige idéer. I det følgende afsnit vil vi beskrive og demonstrere nogle af disse idéer. Vi vil forklare deres oprindelse, hvilke problemer de skulle løse, og hvorfor de blev implementeret eller forkastet i den endelige løsning.

## Brainstorm / Ide Sortering


Lav en omdirigeringsliste	Mindre brug af undersider	Kontaktformular skal fremhæves og gøres mere attraktiv	Trustpilot Banner?	FAQ skal nytænkes og omstruktureres
Bring den vigtigste information frem	Brug billeder på en mere kreative måde	Forside layout skal gennemtænkes	Tydeliggøre hvor der kan findes info om security	
Gøre designet mere luftigt	Hittegods siden & listen mangler information	Kundeservicesiden er svær at finde	FAQ Database?	
Hittegodstabellen er svær at forstå	Hittegods siden skal omtænkes ift. content placering	Tilføj fyldstgørende information til alle sider	FAQ skal integreres på siderne	

## Trustpilot / Banner Idé

En af problematikkerne, vi blev opmærksomme på gennem vores User Stories, var den dårlige score på Google Reviews og Trustpilot. Derfor overvejede vi en komponent, der kunne fungere som et "Calling Card" til at få brugere og rejsende til at dele deres gode oplevelser i lufthavnen, da vi så, at størstedelen af kommentarerne på Trustpilot var negativt anlagt.

MI

**Michael**  
15 anmeldelser



---

Har ventet omkring en time og venter stadig på bagage fra EY061 fra Dubai denne morgen håndteret af Aviator.  
You had one job...  
I kan tage denne bedømmelse med når I evaluerer samarbejdet med Aviator

Vi valgte dog at stoppe med at arbejde på denne idé, da vi videre undersøgte kommentarerne på Trustpilot og opdagede, at mange af dem ikke omhandlede lufthavnens tjenester, men derimod flyselskabernes, handlernes eller restauranternes dårlige service, som Københavns Lufthavn fik skylden for. En komponent, der skulle få brugere til at skrive positive kommentarer på anmeldelsessider, ville derfor ikke have den ønskede effekt.

Senere, i en kort opfølgende samtale med Emma, fandt vi ud af at der havde været overvejelser om at lukke Trustpilot-siden helt på grund af de mange anmeldelser, der ikke havde med lufthavnens tjenester at gøre.



## Omdirigeringsliste/FAQ idé

Komponentet som findes i bunden af 4 ud af 5 undersider i vores løsning startede som to forskellige idéer, som skulle løse hver deres udfordring. For det første ville vi have en overskuelig og effektiv måde at sende de brugere, som er endt det forkerte sted, til det rette sted på cph.dk. For det andet ville vi give svar på nogle af de ofte stillede spørgsmål, som vi vidste, at brugerne havde. I den eksisterende løsning ligger disse spørgsmål og svar som et hav af "slug"-sider på det sekundære domæne, som sandsynligvis skal udsluses. Derfor måtte vi inkorporere denne del på siderne i vores løsning.

Resultatet endte med at være en hybrid mellem de to idéer, som både kan besvare de hyppigste spørgsmål, brugerne har, men også kan pege brugeren videre til andre steder på cph.dk, hvis de er gået forkert. På vores indexside er hovedformålet med komponentet at sende brugerne videre hvis de er gået forkert, hvorimod det på de andre sider er at give yderligere information om undersidens emne.

## Undernavigations Idé

Efter projektets opstart blev vi hurtigt opmærksomme på de store udfordringer, det eksisterende site havde med hensyn til navigation mellem siderne. På det nuværende site er det svært at finde frem til kundeservice, og brugeren bliver desuden sendt væk fra det primære cph.dk-domæne til cph-cem.my.site.com, når de bruger kundeservicesidens navigationsmenu.

For at tydeliggøre for brugeren, at de befinder sig på en af kundeservicesiderne, overvejede vi at implementere en undernavigationsbar. Denne idé blev afprøvet i prototyping-fasen, men blev ikke implementeret, fordi den ville indføre en ny navigationsmetode, som ikke findes andre steder på cph.dk. Dette ville ikke være foreneligt med CPHs visuelle identitet (CVI) og den større kontekst, som kundeservice er en del af.



(Første version af undernavigationen)



(Senere version af undernavigationen)



## Blå Kors Card Idé

Ved besøget hos hittegodsafdelingen blev vi informeret om, at uafhentede genstande blev doneret til Blå Kors. Vi undrede os over, at denne information ikke fremgik nogen steder på hjemmesiden. Derfor besluttede vi at inkludere denne oplysning på vores side.

Denne beslutning blev truffet, da vi mente, det ville øge gennemsigtigheden kundeservice. Ved at informere brugerne om, hvad der sker med uafhentede genstande, skaber vi tillid og viser, at lufthavnen støtter en god sag. Derudover kan vi reducere antallet af genstande, som hittegodsafdelingen skal håndtere. Når brugerne ved, at deres ejendele vil blive doneret til en god sag, hvis de ikke hentes, kan det mindske deres bekymringer og potentielt reducere antallet af henvendelser, omkring mindre værdifulde genstande, til hittegodsafdelingen. Dette kan lette arbejdsbyrden for afdelingens personale og forbedre effektiviteten.



Uafhentede genstande doneres til Blå Kors, som hjælper udsatte børn, unge og voksne i Danmark.



# Komponent idégenerering

Nogle af de komponenter, vi havde brug for, krævede mere dybdegående idéudvikling. For at sikre, at disse komponenter blev så effektive og brugervenlige som muligt, gennemførte vi en omfattende idéudviklingsproces på dem. Denne proces omfattede anvendelsen af flere forskellige metoder, herunder brainstorming, Crazy 8's og skitsering.

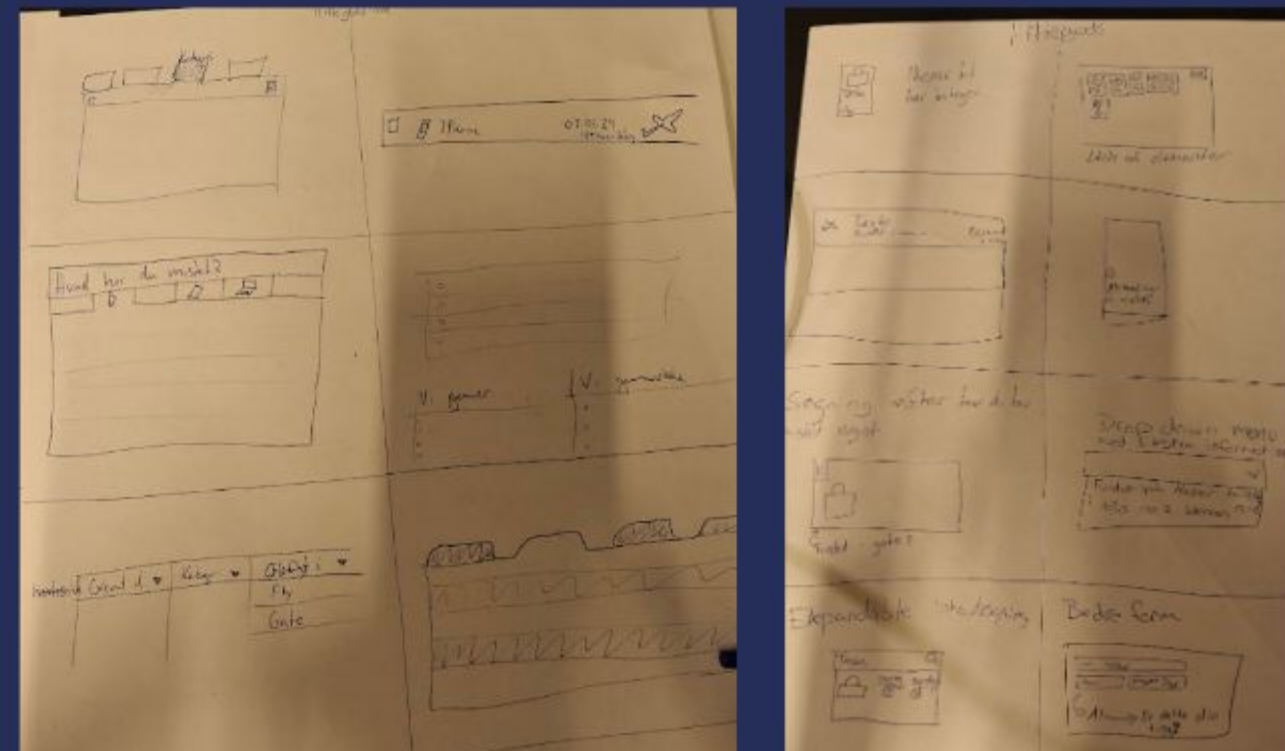
## Hittegodsliste

Et af de områder, der fra projektets start havde vores opmærksomhed, var løsningen af de udfordringer, som hittegodssiden tydeligt havde. Derfor investerede vi betydelig tid og energi i udarbejdelsen af hittegodslisten. Til udviklingen af denne anvendte vi flere idégenereringsmetoder, såsom brainstorm og "Crazy 8s". Senere i processen brugte vi både analoge og digitale værktøjer til skitsering og prototyping. I det vil vi beskrive nogle af de metoder vi brugte i udviklingen af dette komponent.

### Crazy 8s

Crazy 8's<sup>14</sup> er en designmetode, der bruges til at tilskynde deltagerne til at tænke hurtigt og kreativt ved at tegne op til otte forskellige idéer på otte minutter. Målet ved dette er at udforske så mange forskellige løsninger som muligt inden for en kort tidsramme.

Vi brugte dette til at opnå idéer til vores Hittegods Liste komponent og hvordan vi eventuelt kan lave den bedre.



### Skitsering

Ved brug af vores Crazy 8's resultater, dannede vi et grundlag til vores skitsering, som gav os muligheden for at videreudvikle de bedste idéer. Under vores skitserings process skitserede vi selve Hittegods listen, og Hittegods kortet der er på den liste.

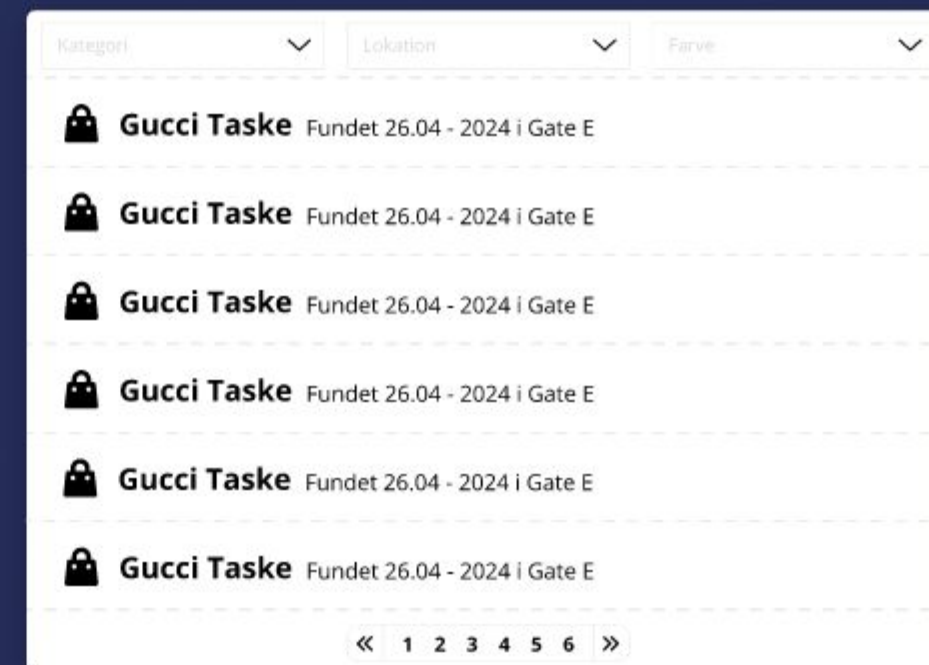


Vi startede ud med at tegne vores ideer i hånden, hvilket tillod os hurtigt at visualisere idéerne fra vores Crazy 8s samlet som en enhed. Vi fokuserede på elementer såsom kategorisering af hittegods, filtreringsmuligheder (Kategori & Lokation), og tydelige visning af fundne genstande med relevante detaljer som dato, lokation og eventuel beskrivelse.



Vi startede ud med at tegne vores ideer i hånden, hvilket tillod os hurtigt at visualisere idéerne fra vores Crazy 8s samlet som en enhed. Vi fokuserede på elementer såsom kategorisering af hittegods, filtreringsmuligheder (Kategori & Lokation), og tydelige visning af fundne genstande med relevante detaljer som dato, lokation og eventuel beskrivelse.

Derefter oversatte vi vores skitser til mere detaljeorienteret prototyper, hvor vi tilføjede ikoner, bedre layout og tags inspireret af sociale netværksapps, som vi efter intern diskussion lavede til fyldestgørende komponenter.



## Kontaktformular

Fra vores kvalitative interview med kundeserviceagent Emma vidste vi, at medarbejderne ønskede, at henvendelserne skete gennem kontaktformularen. Vi vidste også, at flere brugere ringede i stedet for at bruge formularen, fordi de oplevede den som uoverskuelig eller forvirrende.



Den eksisterende formular indeholder 15 kategorier, hvor flere af disse har underkategorier, som også skal vælges. Formularen giver også mulighed for henvendelser om emner, som ikke er inden for kundeserviceområdet, og som de derfor ikke direkte kan hjælpe med. Altså inviterer formularen til henvendelser, som kundeservice ikke kan hjælpe med, hvilket både giver en dårlig oplevelse for brugerne og medarbejderne samt påvirker lufthavnens renommé negativt.

Vi satte os for at skabe en formular, som var mere enkel og overskuelig. Formularen skulle kun indeholde de emner, som rent faktisk kunne afklares af kundeservice. Resultatet kan ses på vores kontaktside.

**Formular på CPHs eksisterende side, kan finde her:**

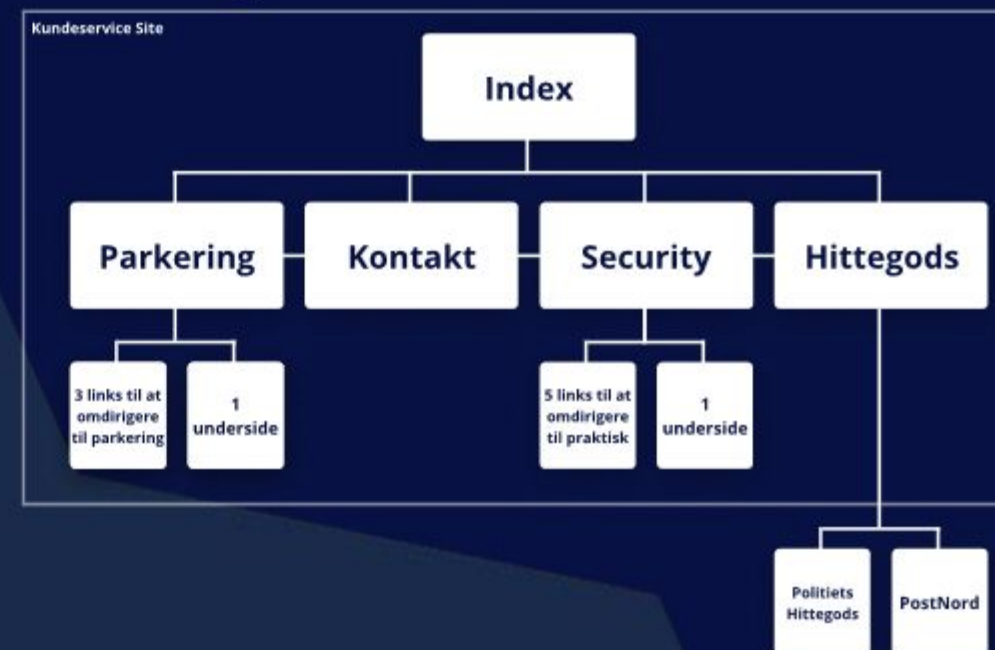
<https://cph-cem.my.site.com/customerservice/s/contactsupport>

## Sitemap for vores webløsning

Ved hjælp af vores Card Sorting-grupperinger og kategorier lavede vi et grundlæggende sitemap. Vores mål var at skabe et sitemap for en webløsning med markant færre undersider og mulighed for at omdirigere brugerne til den rigtige information i stedet for at gentage den flere steder, såsom på kundeservicesiden.

Derfor sigtede vi efter at lave en komponent, der vil skabe nem navigation mellem de forskellige sider og at gøre det let at finde den ønskede information ved at bruge forskellige komponenter på undersiderne: Parkering, Kontakt, Security og Hittegods.

### Vores Webløsning







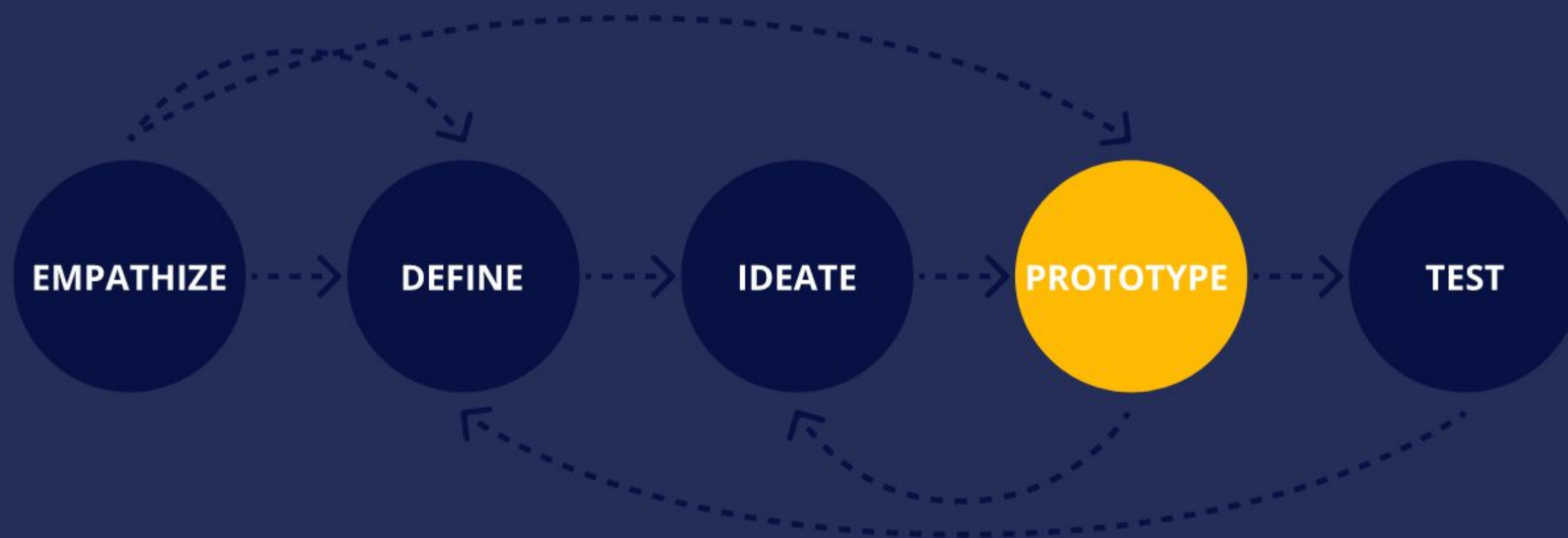
→ **Prototype**



# Prototype

I Prototype-fasen omsættes de udvalgte ideer til en funktionel prototype. Dette kan indebære at skabe prototyper af enkelte funktioner, interfaces eller endda hele systemer.

Prototype-fasen er en vigtig del af processen, der giver mulighed for at afprøve og forfine løsninger inden de implementeres. Ved at skabe prototyper og teste dem, får man indsigt i hvordan brugerne reagerer på løsningen, og kan dermed skabe en mere effektiv og brugervenligt slutprodukt.





# Oversættelse af CVI

Et CVI står for Corporate Visual Identity, hvilket oversættes til virksomhedens visuelle identitet. Det refererer til den visuelle repræsentation af en virksomhed og omfatter forskellige elementer, der tilsammen skaber et sammenhængende og genkendeligt visuelt billede af en virksomhed.

De mest kendte elementer man finder i et CVI er:

**Logo:** Virksomhedens primære visuelle symbol.

**Farvepalette:** De specifikke farver der anvendes i alle visuelle materialer.

**Typografi:** De skrifttyper, der bruges i virksomhedens kommunikation.

**Designprincipper:** Retningslinjer for layout, billeder og grafiske elementer.

Visuelle elementer: Ikoner, mønstre og andre grafiske detaljer.

I vores arbejde har vi fået udleveret CPH Lufthavns CVI [se i filbilag] men da det er markant større end hvad vi har brug for, da det også dækker corporate og partner delen af CPH Lufthavn, valgte vi at oversætte CVI'et til et mere læsbart format, altså et Style Tile.

## Style tile

En Style Tile er et designværktøj, der samler alle de visuelle elementer af et brand på en enkel og overskuelig måde. Dette gør det muligt for os at have en klar og præcis guide til hvordan vi som designere og udviklere kan sikre, at vi bevarer vores kundes visuelle identitet i arbejdet med deres webløsning.

Style Tiles giver os også muligheden for at eksperimentere med forskellige visuelle elementer og se, hvad der fungerer sammen, uden at skulle skabe fuldstændige layouts. Dette giver os en fleksibilitet, der er vigtig, når vi skaber et redesign til en allerede etableret virksomhed som CPH Lufthavn er.

Igennem vores projekt har vores Style Tile været igennem flere iterationer, da det har givet os muligheden for at finpudse og justere designet løbende.

Style tilet kan ses på næste side.





### STYLE TILE EKSEMPEL

## Eksempel menu

Dette er et eksempel på hvordan vores side kunne se ud, udfra CVI og Style Tile.



## Eksempel Tekst til en Headline

En under headline som understøtter teksten over, for at give et mere fyldesgørende indtryk af webside.

### Eksempel på et ofte stillet spørgsmål

Antallet af indcheckede bagagestykker, håndbagage og personlige genstande varierer afhængigt af din billettype, destination og de specifikke regler fastsat af dit flyselskab.

Derfor anbefaler vi, at du læser nærmere på dit flyselskabs hjemmeside for yderligere information. Du kan finde



### Eksempel Card

Engarendende Card Tekst, som giver en ide til hvordan et card på CPH Kundeservice side kunne se ud.

[Learn More](#)



# Wireframe

Da tiden kom til at opstarte Prototyping-fasen, satte vi os for at udforme et sæt wireframes hver, baseret på vores komponent idégenerering. Det viste sig dog, at begge medlemmer, uafhængigt af hinanden, var gået til opgaven med et mere high-fidelity-mindset end planlagt, hvilket resulterede i mere detaljerede forslag end hvad wireframes normalt tilbyder. Vi valgte at omfavne denne metode og besluttede at fortsætte prototyping-fasen ved at designe på komponent-niveau. Dette gav os flere forslag til de fleste komponenter, som vi kunne vælge mellem eller kombinere.

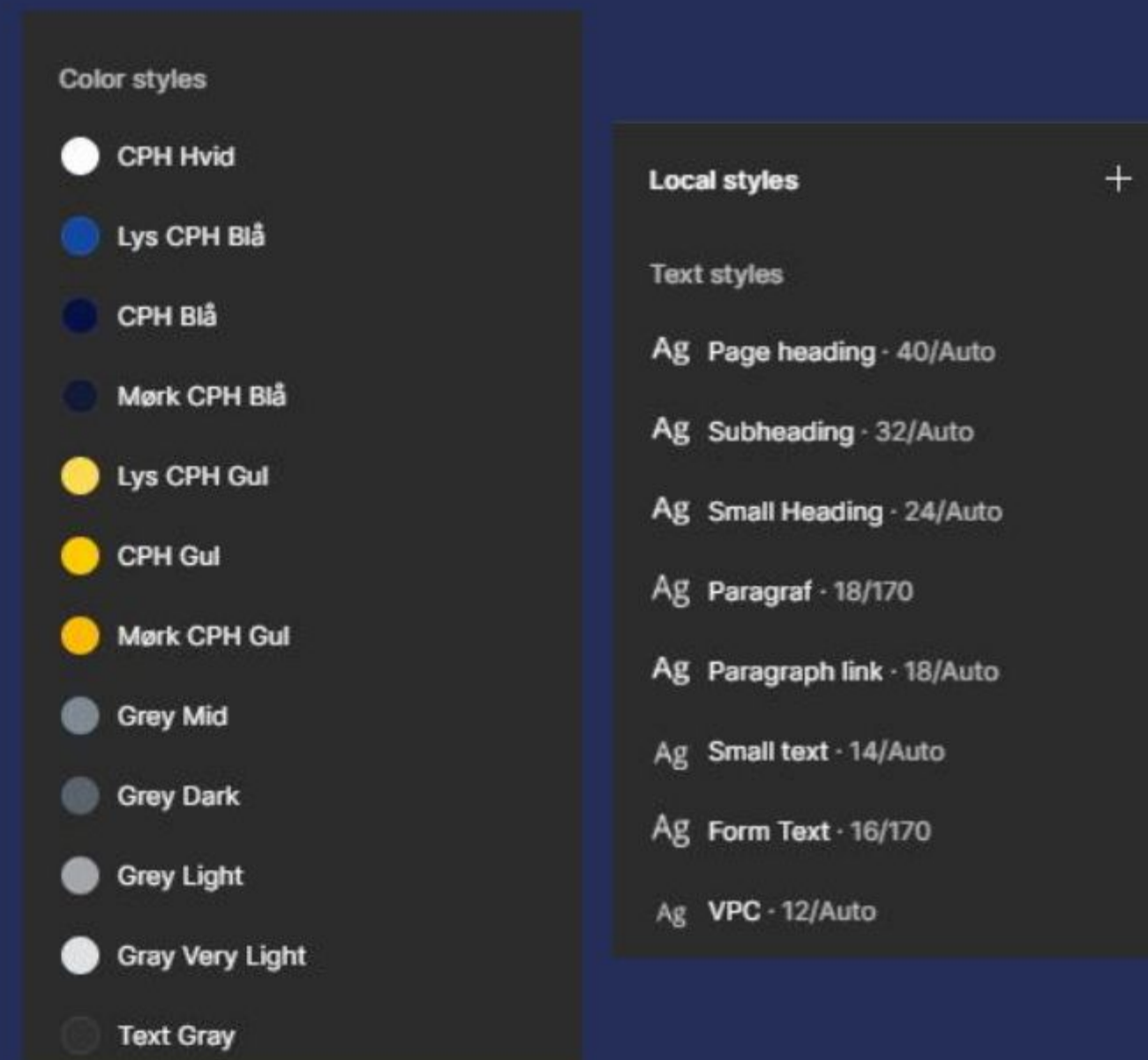


# Mockup

En mockup bruges til at visualisere layoutet af et design uden at bruge tid på at få alle detaljer med. Den bruges til at skabe overblik og sjældent til at vise funktionalitet.

Til at designe vores løsning har vi brugt designredskabet Figma. Med Figma kunne vi skabe alle byggeblokkene til mockuppen ved at gemme de forskellige farver, fonte, og effekter vi havde fra CPHs CVI og gemme dem i projektets lokale "styles". På den måde havde vi begge nem adgang til samme elementer og arbejdede ud fra samme referenceramme.

Fra vores wireframing stod vi med flere udgaver udgaver af en række komponenter, som nu skulle samles på sider og tilpasses, så de passede ind med nærliggende elementer. Fordi vi fra wireframing havde mere færdige komponenter end normalt, var overgangen mellem disse to delprocesser flydende. Komponenterne blev udvalgt og sammensat til hele sider, ved hjælp af inspiration fra vores CVI, style tile og ved at se på den eksisterende hjemmesides løsning. Derefter var vi klar til at gå videre til at skabe en prototype.





# Prototype

En prototype er en tidlig version af et produkt, som bygges for at teste og validere et design. I webudvikling er en prototype en interaktiv model af en hjemmeside, der simulerer, hvordan den endelige version vil fungere og se ud. En prototype kan omfatte hele løsningen eller specifikke komponenter.

I vores proces var vejen fra mockup til vores Figma-prototype ikke lang. Processen involverede at foretage de sidste beslutninger, tilføje detaljer og lave forbindelser mellem de forskellige sider og elementer, samt tilknytte pop-up modaler.

Selvom vi har lavet en forholdsvis detaljeret prototype i Figma, stoppede prototyping ikke der. I webudvikling kan man fortsætte med at iterere og forbedre design og funktionalitet gennem hele udviklingsprocessen. Dette kan ses som en form for fortsat prototyping, hvor vi løbende tester, evaluerer og tilpasser løsningen baseret på feedback, nye indsigter og opdagelser undervejs.

På grund af vores præference til kodning vurderede vi, at det ville være ineffektivt at udvikle dele som f.eks. animationer i Figma. Derfor besluttede vi, at yderligere iterationer ville foregå gennem kodning. Ved at bruge kodning som en prototyping-metode kunne vi hurtigt implementere, teste og justere animationer og interaktioner direkte i det format, som den endelige løsning skal have. Dette gav os mulighed for at eksperimentere og forbedre vores design på en mere effektiv og præcis måde.





# Afgrænsning af løsning

For at læsere af denne rapport nemmere kan forstå vores webløsning, vil vi her udpege, hvilke dele af hjemmesiden der ikke skal opfattes som funktionelle dele af løsningen, men enten er omgivende strukturer, som skaber kontekst for løsningen, eller bevidst er gjort ufunktionelle.

For at vise vores løsning i realistiske omgivelser har vi kodet footeren og headeren samt deres navigation som en nogenlunde efterligning af de oprindelige. Disse komponenter er ikke noget vi har mandat til at ændre, men de er nødvendige for det visuelle udtryk på vores website. Af den grund er de fleste links i disse elementer tomme, da vi ikke ønsker at føre brugeren væk fra løsningen. De links, der bør pege på vores undersider, gør dette.

En del af vores løsning handler om at sende de brugere, der er endt forkert, til det korrekte sted. Dette gøres for eksempel på vores parkeringside, hvor et link fører til en informativ side om handicapparkering på cph.dk. Det samme gælder for mange af de links, der findes i vores FAQ-sektioner.







**→ Teknisk dokumentation**



# Teknisk dokumentation

## Valg af framework

Da tiden kom til at vælge framework, var der flere faktorer, som spillede ind i vores overvejelser. Vi havde til en vis grad samme referenceramme og kendskab til teknologier fra vores uddannelse. Dette betød, at vi havde færdigheder inden for de samme frameworks, men i flere tilfælde forskellige oplevelser og erfaringer med dem.

En vigtig faktor i udvælgelsen var at vælge et framework, som vi ønskede at dygtiggøre os inden for, med henblik på fremtidigt arbejde.

Vi overvejede især følgende frameworks:



### Next.js

Dette har været omdrejningspunktet for undervisningen på vores 3. semesters valgfag.



### Astro

Dette framework har vi arbejdet med på 2. semester og udviklet flere projekter i.



### Solid.js

Et teammedlem har brugt dette framework i sin praktikperiode og til tidligere projekter.

Det var ligeledes vigtigt for os at vælge en teknologi, som var velegnet til at løse den specifikke opgave, vi stod overfor.

Med disse overvejelser i mente besluttede vi at bygge projektet i Astro. Astro tilbyder muligheden for at integrere andre frameworks, hvilket vi har valgt at gøre med React.js, for at have større råderum, når det kommer til de mere dynamiske funktioniteter i vores løsning.



# Astro

Astro er et frontend-framework, som følger en komponentbaseret, der tillader modulær kodestruktur. Det er designet til at kunne understøtte og integrere andre forskellige frameworks som React, Vue, Svelte og andre. Astro i sig selv fokuserer på at levere statiske hjemmesider ved at generere mest muligt statisk HTML.

Frameworket giver også fleksibilitet til at tilføje og administrere forskellige datakilder og tilpasse applikationer efter behov.

I vores projekt har vi valgt at bruge Astro på grund af dets enkelhed og overskuelighed samt fordi den komponentbaserede tilgang gør at man effektivt kan arbejde med det i et team. Astros komponentbaserede arkitektur har gjort det muligt for os at opbygge vores applikation i komponenter, hvilket gør vedligeholdelse og opdateringer meget nemmere. Derudover gør Astro det også hurtigt at udvikle, pga. dets letforståelige syntaks og samling af komponentets HTML, CSS og JS i samme fil.

Astro er også kendt for sin gode performance, sammenlignet med andre frameworks, særligt når det kommer til statiske sider, hvor Astros kan levere sider, der loader hurtigt og dermed forbedrer brugeroplevelsen.

Dog har Astro også nogle mangler. Det egner sig bedst til at generere hjemmesider og kan være mindre ideelt for mere dynamiske sider. Astro er nyt og har derfor et mindre npm-økosystem sammenlignet med større og ældre frameworks som React eller Vue.

I dette projekt har vi valgt at integrere React.js i Astro, for at have adgang til Astros simplicitet og Reacts interaktionsfunktionalitet.

Astros mulighed for at integrere andre frameworks gør at vi på de forskellige sider kan importere og blande React-komponenter (.jsx-filer) med Astro-komponenter (.astro-filer).

```
Hero.astro
---
import { Image } from "astro:assets";

const { heroTitle, heroDescription, heroImage, heroImageAlt } = Astro.props;
---

<section class="flex justify-between max-h-[300px]">
  <div
    class="flex justify-end items-center p-4 w-full sm:w-1/2 bg-grey-lighter"
  >
    <div class="max-w-[500px] w-full">
      <a class="text-light-blue font-bold" href="/">KUNDESERVICE</a>
      <h1 class="font-bold text-cph-blue text-4 sm:text-3 md:text-2">
        {heroTitle}
      </h1>
      <p class="text-cph-blue pe-4">{heroDescription}</p>
    </div>
  </div>
</div>
```



# React

React.js er et populært JavaScript-bibliotek til udvikling af brugergrænseflader i webapplikationer. Det er designet til at være fleksibelt, effektivt og genanvendeligt. Med React.js kan udviklere opbygge komponentbaserede brugergrænseflader, der gør det nemt at styre tilstanden (state) af applikationen og opdatere visningen, når der forekommer ændringer.

I projektet har vi haft gavn af at bruge Reacts muligheder for at lave interaktive komponenter, som reagerer på brugerinput. React er anvendt til nogle af de komponenter, som kræver høj grad af brugerinteraktion eller 'conditional rendering', heriblandt sidens Header, Hittegodsliste og FAQ samt nogle af deres underkomponenter.

```
package.json

{
  "name": "cph",
  "type": "module",
  "version": "0.0.1",
  "scripts": {
    "dev": "astro dev",
    "start": "astro dev",
    "build": "astro build",
    "preview": "astro preview",
    "astro": "astro"
  },
  "dependencies": {
    "@astrojs/react": "^3.3.2",
    "@astrojs/tailwind": "^5.1.0",
```



# CSS styling

Da det kom til valg af metode til anvendelse af CSS, var der forskellige præferencer i teamet. En af teammedlemmerne foretrak at bruge vanilla CSS, mens et andet medlem foretrak Tailwind CSS. Efter en del diskussion kunne vi ikke finde et stærkt argument for at tvinge nogen til at arbejde uden for deres komfortzone eller præference. Vi besluttede derfor at tillade brugen af begge metoder for at sikre, at alle kunne arbejde effektivt og komfortabelt.

Selvfølgelig er der nogle overvejelser ved at kombinere disse to tilgange. For eksempel vil der være en smule ekstra 'page weight' fra Tailwind CSS, som ikke vil blive udnyttet fuldt ud, da vi også anvender vanilla CSS. Dette kunne potentielt føre til en minimal stigning i side indlæsningstiden. Dog vurderede vi, at fordelene ved at lade teammedlemmerne arbejde med deres foretrukne værktøjer opvejede denne ulempe. På den måde kunne vi også understøtte de mål, som vi individuelt havde for vores egen læring. Fordi begge teammedlemmer har en vis forståelse for begge metoder, skabte kombinationen ikke udfordringer, men gav skabte, på den ene side kendskab til Tailwind, og på den anden side erfaring med at inkorporere f.eks. custom properties i Tailwind, således at de samme properties bruges af begge medlemmer.

## Tailwind CSS

Tailwind CSS er et CSS-framework som har en "utility-first" arbejdsproces, idet det forenkler designprocessen ved at tilbyde et bredt udvalg af foruddefinerede klasser. I stedet for at skrive brugerdefinerede CSS-stilarter kan udviklere bruge Tailwinds klasser direkte i HTML for hurtigt at skabe et responsive og moderne design.

I vores brug af Tailwind har vi oplevet flere fordele. Først og fremmest har vi kunnet definere vores egne tekststørrelser, fonte og farver i Tailwind. Dette har gjort det muligt for os at opretholde en konsekvent stil på tværs af vores projekt, hvilket er essentielt for at skabe en sammenhængende visuel identitet, samt at opretholde CPH lufthavns CVI. Ved at kunne tilpasse disse elementer efter vores eget behov, har vi sikret at vores design og komponenter er tilpasset CPHs brand image.



Derudover har Tailwind gjort det let at skabe ensartethed i vores komponenter, og samtidigt forenkle processen i at gøre komponenterne skalerbare til alle enheders skærmstørrelser. Med de mange foruddefinerede klasser har vi hurtigt kunnet implementere responsive design principper, som betyder at vores webløsning ser pæn ud på alle enheder uanset skærmstørrelse.

På billede til højre ses det hvordan vi har sat vores custom properties op i tailwind. Dette tog vi fra CPH Lufthavns CVI, og gav dem passende navne.

Dette betød at vores standard CSS variabler bliver brugt på Tailwind CSS, hvilket gør at vi bruger de samme font størrelser og farver, selvom vi anvender styling ved brug af to forskellige metoder.

En af de største ulemper ved Tailwind, er hvordan man hurtigt kan give en komponent en masse overflødige klasser som ikke gør noget, da man bare kan "smide" klasser på indtil at det man laver virker.

```
tailwind.config.js

theme: {
  colors: {
    "dark-blue": "var(--dark-blue)",
    "cph-blue": "var(--cph-blue)",
    "light-blue": "var(--light-blue)",
    "dark-yellow": "var(--dark-yellow)",
    "cph-yellow": "var(--cph-yellow)",
    "light-yellow": "var(--light-yellow)",
    "cph-white": "var(--cph-white)",
    "grey-lighter": "var(--grey-lighter)",
    "grey-light": "var(--grey-light)",
    "grey-mid": "var(--grey-mid)",
    "grey-dark": "var(--grey-dark)",
    "grey-text": "var(--grey-text)",
  },
  fontSize: {
    1: "var(--fs-1)",
    2: "var(--fs-2)",
    3: "var(--fs-3)",
    4: "var(--fs-4)",
    5: "var(--fs-5)",
    6: "var(--fs-6)",
```



# NPM-pakker

En af de store fordele ved at anvende populære frameworks som React og Astro er, at de mange tidligere brugere sandsynligvis har stødt på de samme udfordringer, som du nu står overfor. Dette fællesskab af brugere fører til udviklingen af biblioteker og komponenter, der er designet til at løse de mest almindelige problemer og mangler ved disse frameworks. Med andre ord, hvis du støder på en udfordring, er chancen stor for, at nogen allerede har fundet en løsning, og at de har bygget et værktøj eller komponent, hvori løsningen kan deles. Dette sparer tid og gør udviklingsprocessen mere effektiv, da du kan drage nytte af eksisterende ressourcer.

Det har vi også udnyttet i vores udvikling og vi vil i det følgende beskrive nogle af de npm-installs, der har hjulpet os med vores løsning.

## React Markdown

React-Markdown er en populær npm-pakke, der gør det muligt at renderere markdown-indhold som React-komponenter. Ved at bruge React-Markdown kan vi nemt integrere markdown-tekst i vores React-komponenter og få den gengivet som HTML

Måden React-Markdown virker på er ved at bruge en parser til at konvertere markdown-tekst til et abstrakt syntakstræ (AST). Den mest almindelige parser, som React-Markdown anvender, er "Remark", som er et populært markdown-parsing-bibliotek.

Når markdown-teksten er blevet parset til en AST, traverserer React-Markdown dette træ og konverterer hver node i træet til en tilsvarende React-komponent. For eksempel, når React-Markdown støder på en heading-node (f.eks. "## Hej"), konverterer denne node til en "<h2>" react-komponent.

Det gør den med alle Markdown elementer, som gør at vi kan style vores FAQ svar ved brug af Markdown language.



## React-transition-group

I kodningen af burgermenuens navigation anvender vi conditional rendering til at vise de forskellige lister under hvert menupunkt. Dette skabte en udfordring i forhold til den tilhørende animation, der er knyttet til åbne/lukke-funktionen, da conditional rendering tilføjer og fjerner elementer fra DOM'en. På grund af dette er det ikke muligt at styre animationen via CSS alene, da CSS ikke kan animere elementer, der dynamisk fjernes eller tilføjes til DOM'en.

Løsningen fandtes i React-biblioteket React-Transition-Group, som håndterer elementer, når de tilføjes eller fjernes fra DOM'en, og integrerer en transition. Biblioteket sørger for, at et angivet element, i det øjeblik det indtræder i DOM'en, starter med opacity: 0; og height: 0; og derfra animerer til fuld synlighed og højde.

Nedenfor ses et eksempel på koden:

```
React-transition-group

<li className={"menu-item " + (liThreeActive ? "active" : "")} onClick={handleLiThreeClick}>
  <p>Parkering</p>
  <CSSTransition in={liThreeActive} timeout={300} classNames="inner-list" unmountOnExit>
    <InnerList itemKey="parkeringContent" />
  </CSSTransition>
</li>
```

I dette eksempel anvendes CSSTransition-komponenten. Prop'en 'in' styrer, hvilken state visningen er afhængig af, 'timeout' angiver transitionens varighed, og 'unmountOnExit' fjerner elementet fra DOM'en, når det ikke længere er synligt. Inde i CSSTransition-komponenten findes 'InnerList', som er det JSX-komponent, vi selv har bygget, og som skal vises dynamisk.

Processen med at genskabe den header og navigation, der allerede eksisterer på cph.dk, har været en større udfordring end forventet, men en lærerig udfordring. Samtidig har vi måttet afveje, hvor meget tid vi skulle bruge på denne opgave, da den ikke direkte er en del af projektet, men snarere et omgivende element, der understøtter konteksten. Derfor var det kærkomment, at denne npm-pakke kunne løse en del af opgaven nogenlunde gnidningsfrit.



### CLSX

CLSX er en lille, effektiv npm-pakke, der gør det nemt at sammensætte betingede CSS-klasser.

```
buttonExample.js

import clsx from 'clsx';

const Button = ({ isPrimary, isDisabled }) => {
  const className = clsx({
    'btn': true,
    'btn-primary': isPrimary,
    'btn-disabled': isDisabled
  });

  return <button className={className}>Click me</button>;
};
```

I dette eksempel vil clsx hjælpe med at tildele de korrekte CSS-klasser baseret på "isPrimary" og "isDisabled" props, hvilket gør koden mere læsbar og vedligeholdelsesvenlig.





# Eksempler fra koden

For at give et indblik i vores process med at kode hjemmesiden, vil vi i det følgende beskrive to komponenter og deres kodemæssige opbygning. Vi vil forsøge at beskrive de udfordringer vi stødte på i kodningen af disse og forklare hvordan de blev løst i konteksten af det valgte framework.

## Header

En af de større kodemæssige udfordringer var, sjovt nok, ikke en komponent vi selv havde designet, men at genskabe headeren fra cph.dk. Selvom headeren ikke har den store betydning for vores løsning, da headeren på cph.dk ikke engang indeholder et link til kundeservicesiden, var den stadig en stor del af siden udtryk og måtte derfor genskabes. I vores løsning har vi tilføjet et link til kundeservicesiden under menupunktet "Praktisk".

Den kodemæssige udfordring lå i den dropdown-menu, som åbner når musen føres over menupunkterne, og det at vise forskellige undermenuer baseret på hvilket punkt der bliver "hovered" over.

Løsningen blev at bygge dropdown-menuen som komponent, og lægge det i hvert menupunkt, men modtage en prop [1], som fortæller komponentet, hvilken data den skal vise.

Hvert menupunkt har "event handlers" tilknyttet, som eksekverer en funktion ved når musen føres over elementet, ved brug af Reacts onMouseEnter-event [2].

Dette starter så funktionen mouseEnter [3], som så aktiverer "statet" showDropDown, med den tilsvarende item [4].

Dermed renderes DropDown-komponentet med prop'en item="Flyinformation" [5].

```
Header.jsx
<ul className="upper-nav">
  <li className="upper-li" onMouseEnter={() => mouseEnter("Flyinformation")} onMouseLeave={mouseLeave}>
    <a href="">Flyinformation</a>
    {showDropDown === "Flyinformation" &&
      <DropDown item="Flyinformation" onMouseEnter={dropDownEnter} onMouseLeave={dropDownLeave} />
    }
  </li>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

```
Header.jsx
[3]const mouseEnter = (item) => {
  setShowDropDown(item);[4]
  clearTimeout(timeoutRef.current);
};

const mouseLeave = () => {
  timeoutRef.current = setTimeout(() => {
    setShowDropDown(null);
  }, 300);
};
```



I dropdown-komponentet bruges javascript-funktionen "map()" til at vise den korrekte data.

Map-funktionen "looper" over et array, behandler arrayets elementer og returnerer et nyt array. I første map-funktion[1] returneres et <a> for hver item i arrayet "content.links", som i hver item har en "text" og en "href", som bruges til at udfylde <a>-tagget.

Den anden map-funktion itererer over arrayet "content.lists", som er et array af arrays, altså en liste af lister. Den anden map-funktion[2] returnerer en <ul> for hver af disse arrays.

For at hver af disse listers items kan blive vist, skal den tredje map-funktion[3] i brug. Den sørger for at hvert objekt inde i disse lister vises som et <a> inde i et <li>.

Dataen befinder sig i forskellige objekter[4] inde i komponenten.

```
DropDown.jsx
<nav className="dropdown-menu" onMouseEnter={() => {}} onMouseLeave={() => {}}>
  <div className="dropdown-text">
    <h3>{content.title}</h3>
    <p>{content.description}</p>
    <div className="link-container">
      [1]{content.links.map((link, index) => (
        <a key={index} href={link.href}>
          <span>&#x2192;</span> {link.text}
        </a>
      ))}
    </div>
  </div>
  [2]{content.lists.map((list, index) => (
    <ul className="dropdown-list" key={index}>
      [3]{list.map((item, i) => (
        <li key={i}>
          <a href={item.href}>{item.text}</a>
        </li>
      ))}
    </ul>
  ))}
</nav>
```

```
const flyinformationContent = { [4]
  title: "Flyinformation",
  description: "Find nemt alle afgange og ankomster og få et overblik over flyselskaber.",
  links: [{ text: "Find dit fly", href: "" }],
  lists: [
    [
      { text: "Flyinformation", href: "" },
      { text: "Afgange", href: "" },
      { text: "Ankomster", href: "" },
      { text: "Transfer", href: "" },
      { text: "Indenrigsflyvninger", href: "" },
      { text: "Destinationer", href: "" },
    ],
    [
      { text: "Selskaber/Partnere", href: "" },
      { text: "Flyselskaber", href: "" },
      { text: "Handlingselskaber", href: "" },
    ],
    [
      { text: "", href: "" },
      { text: "", href: "" },
      { text: "", href: "" },
    ]
  ]
}
```



## Pagination Komponent

Denne kodedel er en del af en komponent der er skrevet i React, som bruges i vores komponent der hedder "Pagination.jsx" der bliver brugt til at håndtere de forskellige sider af vores Hittegods Liste. Pagination er processen at opdele en lang liste af elementer i mindre dele eller sider, som brugerne kan navigere mellem.

**1** - Den funktionelle komponent kalder "Pagination", som tager fire props (egenskaber) som input:

**currentPage:** Den nuværende side som brugeren ser.

**totalCount:** Det totale antal elementer der skal vises.

**pageSize:** Antallet af elementer der vises per side.

**onPageChange:** En funktion der bliver kaldt, når brugeren skifter side.

**2** - Her beregnes det totale antal sider "totalCount" som bliver divideret med "pageSize" og resultatet rundes op til nærmeste hele tal med "Math.ceil" fordi der ikke kan være halve sider.

**3** - Hvis der kun er én side, returnerer komponenten "null", hvilket betyder at der ikke vises nogen pagination, da der ikke er behov for at skifte mellem siderne.

**4** - Til sidst oprettes en liste (array) af sider. En tom liste "pages" initialiseres og et "for"-loop bruges til at fylde denne liste med tal fra 1 til "totalPages"

```
Pagination.jsx
1 const Pagination = ({ currentPage, totalCount, pageSize, onPageChange }) => {
2   const totalPages = Math.ceil(totalCount / pageSize);
3   if (totalPages === 1) return null;
4   const pages = [];
   for (let i = 1; i <= totalPages; i++) {
     pages.push(i);
   }
}
```

Resten af kode der ikke er vist her, bruges til at renderere disse sideindekser som en klikbar søjle i bunden af Hittegodslisten, ved brug af html og Tailwind CSS.



# Supabase

I vores projekt har vi valgt at bruge Supabase til flere formål, herunder håndtering af vores hittegods liste og reservationssystem.

```
getItems.js

export async function getItems(sortBy, filterBy, page = 1, pageSize = 6) {
  const from = (page - 1) * pageSize;
  const to = page * pageSize - 1;

  const call = supabase
    .from("hittegodslist")
    .select("*", { count: "exact" })
    .range(from, to);

  if (sortBy) {
    call.order(sortBy, { ascending: true });
  }

  if (filterBy) {
    call.filter("item_type", "eq", filterBy);
  }

  return await call;
}
```

Supabase er en open-source backend-database platform, der tilbyder et komplet sæt værktøjer og tjenester til udvikling af moderne applikationer. Den er baseret på PostgreSQL-database og giver udviklere mulighed for at oprette, administrere og skalere databaser hurtigt og nemt. Supabase tilbyder også realtids funktionalitet ved hjælp af WebSocket-protokollen, hvilket gør det muligt for applikationer at opdatere data øjeblikkeligt uden behov for manuelle opdateringer.

## Hittegods Liste

Vi bruger Supabase til at administrere vores hittegods liste, ved at have alle de fundne genstande i databasen hvor hver genstand indeholder information såsom; navn på genstanden, type, placering og dato. Supabase giver mulighed for at have realtidsopdateringer, så hvis en genstand bliver tilføjet til Hittegods Listen, kan det ses på vores webside efter genindlæsning af siden.

## Reservationssystem

Til reservationssystemet har vi alle mails og deres reservationsnumre liggende så vi kan tjekke den information op mod hvad brugeren indsætter i vores input felter på websiden. Hvis informationen er korrekt, bliver der vist en visuel bekræftelse på siden.



# Source control

Fra den nyligt overståede arbejdspraktik havde gruppemedlemmerne nyvundet erfaring med source control i en professionel sammenhæng. At opleve, i praktikken, at samarbejde og source control foregår på en lidt mere stringent måde på den professionelle arbejdsplads, var ikke nogen overraskelse. Men det var stadig meget kontrastfyldt, da den forudgående erfaring var et samarbejde mellem en gruppe studerende, som lige havde lært hvad Git og GitHub var.

De nye metoder ville vi implementere vores afsluttende projekt for at opnå en bedre struktur, mere sikkerhed omkring vores kode samt større samhørighed i teamet.

Omdrejningspunktet for vores metode var at anvende Pull Requests (PR). En PR er en forespørgsel om den branch man har arbejdet i kan godkendes og 'merges' ind i main-branchen. I vores tilfælde har det andet gruppemedlem så skullet gennemlæse, eventuelt teste, og derefter enten bede om ændringer eller godkende Pull Requesten.

I projektet har vi anvendt en trunk-based development-tilgang, hvilket vil sige at vi fokuserer på at implementere mindre features eller ændringer ad gangen i vores branches og stræber efter at 'merge' dem hurtigt ind i main-branchen.

Ved at implementere små ændringer ad gangen undgår vi 'døde' branches med forældet kode, som aldrig bliver en del af projektet.

Som en yderligere del af metoden, er det vigtigt at hver branch/PR deles op i meningsfulde commits med afgrænset indhold, som forholder sig til et område ad gangen f.eks. kodning af et enkelt komponent. Til hvert commit skal der følge en commit-besked, som kort fortæller hvad der implementeres. Alt dette hjælper med at skabe et godt overblik for det teammedlem som skal gennemse PR'en.

Ved at lade al koden blive gransket ved hjælp af en PR, sikrer vi også at teammedlemmerne har mere kendskab til den kode, som det andet medlem har skrevet, og vi øger dermed forståelsesniveauet af den samlede kodebase. Det giver også mulighed for at give hinanden feedback på koden, samt et forum for faglig diskussion.

For at anvende denne metode, er det nødvendigt med et lidt mere indgående kendskab til Git-kommandoer end de allermest gængse. I arbejdet med dette projekt er vi blevet mere fortrolige med følgende kommandoer.



### **\$ git push --force:**

Tvinger de nuværende ændringer til at erstatte den remote branch i repository, selvom den er forskellig fra den lokale branch.

### **\$ git commit --amend (--no-edit):**

Tilføjer 'stagede' ændringer til det seneste commit. Med --no-edit-tilføjjelsen gemmes den eksisterende commit-besked uden at åbne redigeringstilstand.

### **\$ git rebase <branch-name>:**

Bruges til at omorganisere commits, redigere commit-beskeder osv. i forhold til den angivne branch. I praksis til at få branchens historik til at passe til den branch (oftest main) den skal rebases på, så den kan merges uden konflikter.

### **\$ git cherry-pick <commit-id>:**

Trækker specifikke commits, baseret på deres commit-id, ind i den aktuelle branch.

### **\$ git checkout <branch-name>:**

Skifter til den angivne branch.

### **\$ git checkout -b <new-branch-name>:**

Opretter en ny branch og skifter til den på én gang.

### **\$ git status:**

Viser status omkring ændrede, 'stagede' og 'untracked' filer.

### **\$ git push -u origin <branch-name>:**

Pusher den lokale branch til remote GitHub repository.

Foruden de beskrevne fordele, var formålet med at anvende disse git/ source control-strategier også at tilvænne os en professionel metode, for at det vil komme os til gavn i vores kommende arbejdsliv.

Anvendelsen af metoden gik rimelig gnidningsfrit og over forventning. Forudgående havde vi en mistanke om at det ville være tidskrævende, men det er ikke vores oplevelse at det har været det.



# Næste iteration

Gennem dette projekt er der enkelte features, som enten har været planlagt eller oppe til overvejelse, men som ikke er blevet implementeret i denne iteration.

### **Sprog Toggle:**

Cph.dk har en sprogtoggle, der skifter sproget mellem dansk og engelsk. Undervejs i projektet var det planlagt også at implementere denne funktion i vores løsning. Vi valgte dog at nedprioritere denne feature, da den primært ville medføre en stor mængde mekanisk oversættelsesarbejde. Dette ville være en naturlig opgave til den næste iteration af løsningen. Havde det ikke været et skoleprojekt, ville opgaven have indebåret at integrere cph.dks nuværende sprog toggle-funktionalitet på siderne.

### **Brødkrummesti:**

Gennem vores research fandt vi ud af at et af de grundlæggende udfordringer cph.dk har, er den uklare navigation mellem siderne. Til at løse dette har vi overvejet implementering af en brødkrummesti for at tydeliggøre for brugeren, hvor de befinder sig, samt give dem en bekvem måde at navigere baglæns på.

Denne feature blev dog skåret fra designprocessen, da vi vurderede, at den ikke var forenelig med CPHs nuværende CVI, samt at det i højere grad ville give mening at implementere den sitewide.

### **Kontaktformular:**

Kontaktformularen, som findes i vores løsning, skal mest af alt ses som en visuel forbedring af den eksisterende, hvor vi har indsnævret mulighederne for valg af emne til de vigtigste. Gennem interview med en medarbejder lærte vi, at den nuværende formular var fyldt med overflødige muligheder. Vi har derfor fokuseret på at gøre formularen mere simpel og overskuelig. Da vi allerede to andre steder på sitet forbinder frontenden til en database, vurderede vi, at det var overflødt at bruge tid på at gøre det endnu en gang.

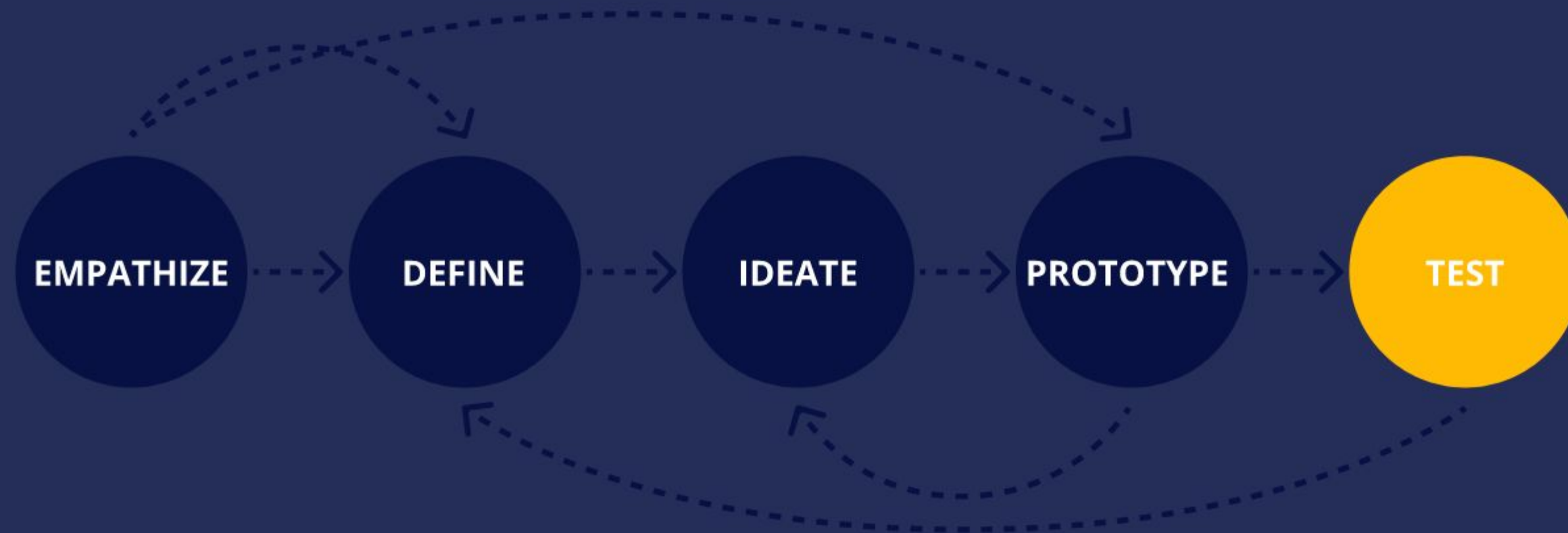






# Test

I Test-fasen evalueres prototyperne med brugere for at validere deres effektivitet og identificere eventuelle problemer eller mangler. Denne fase hjælper med at sikre, at løsningen imødekommer brugernes behov og er af høj kvalitet. Gennem testning vil det blive tydeligt, om der er behov for yderligere iterationer og forbedringer.





# Lighthouse

Lighthouse er et open-source-værktøj udviklet af Google, som vurderer ydeevnen, tilgængeligheden, bedste praksis og søgemaskineoptimering (SEO) for websider. Det udarbejder en detaljeret rapport med handlingsrettede indsigter og anbefalinger, der hjælper med at forbedre den overordnede samlede kvalitet og ydeevne af en hjemmeside.

## Vores løsning

For at teste vores webbløsnings forside, udførte vi en ny Lighthouse test for at tjekke om vores side lever op til de krav vi har sat for os selv, og de standard krav der er inden for accessibility og best practices.

Vores test viste et generelt godt resultat, men ved at dykke dybere og se hvad fejlene er, kan vi se at vi kan opnå et bedre resultat ved at lave små ændringer på vores side. De største fejl på siden omhandlede SEO, da vi havde glemt at tilføje en deskriptiv tekst til vores meta information, dette er vigtige, da Google bruger disse tekster til at vise mere nøjagtige søgeresultater i deres søgemaskine.

**100** Accessibility

**100** Best Practices

Derudover var performance scoren kun 88, da at der ikke var brugt Astro's <Image> tag på alle billeder hvor det var muligt, og der var stort layout shift på nogle af siderne. Dette prøvede vi at fikse, og endte på en score på 90, da vi ikke kunne optimere alle punkterne grundet det vil kræve større ændringer i vores kode infrastruktur.

**88**

## Performance

- Avoid large layout shifts
- Properly size images
- Eliminate render-blocking resources

**90**

## SEO

- Document does not have a meta description

## Vores optimeret løsning

**90**

Performance

**98**

Accessibility

**100**

Best Practices

**100**

SEO



# Page Weight

Page Weight refererer til den samlede størrelse af en webside, herunder alle dens ressourcer såsom HTML, CSS, JavaScript, billeder og andre mediefiler. Det er en måling af mængden af data, der skal downloades af en brugers browser for at vise siden. Page Weight har direkte indflydelse på indlæsningstiden og ydeevnen af en hjemmeside. Optimering af page weight indebærer at reducere filstørrelser, komprimere billeder, minimere kode og anvende effektive leveringsteknikker som caching og indholdsleveringsnetværk.

Vi har derfor testet vores webløsning ved brug af Page Weight. Dette viser os, hvor lang tid siderne er om at loade indholdet, samt hvor meget data der bliver overført fra websiden til brugeren. Vores resultater viser en loading tid mellem 0.5 sekunder til 1.5 sekunder, hvilket vi mener er et acceptabelt resultat, når man vejer det op med den nuværende index side, som tager op mod 6 sekunder at loade.

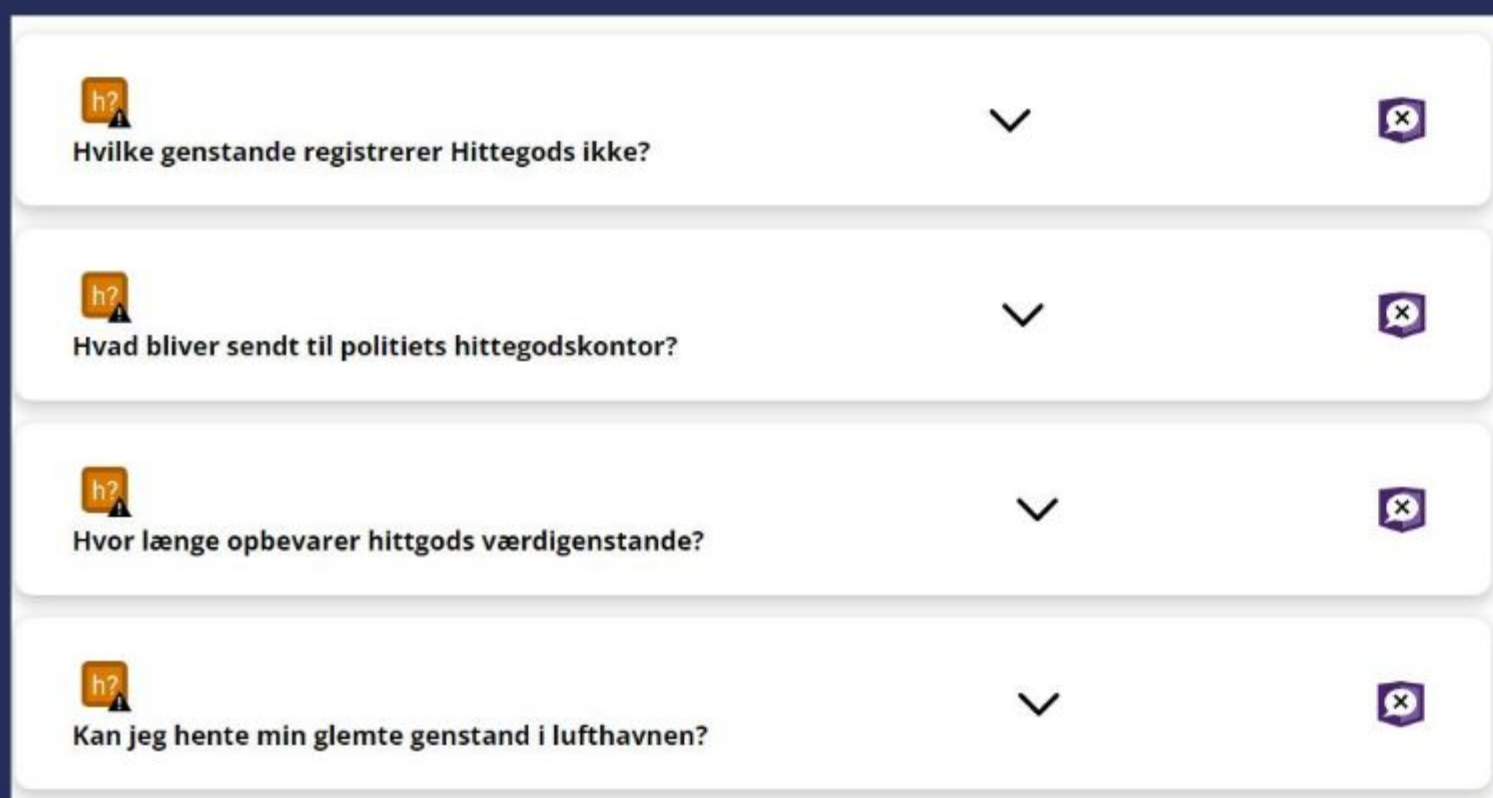
Måden vi har holdt datamængden nede, er ved at have konverteret vores billeder til .webp fil formater, og undgå brug af videoer, da de generelt er kendt for at være tunge i filstørrelsen.





# WAVE

WAVE Web Accessibility Evaluation Tool<sup>15</sup> er et værktøj udviklet af WebAIM, som hjælper med at evaluere tilgængeligheden af websider. WAVE analyserer en side for tilgængelighedsproblemer og genererer en visuel rapport med markeringer og forklaringer af eventuelle problemer. Dette gør det muligt for udviklere at identificere og rette fejl, hvilket forbedrer brugeroplevelsen for personer med handicap og sikrer at tilgængelighedsstandarderne er overholdt.



Ved brug af WAVE Web Accessibility Evaluation Tool fandt vi flere små mangler som let vil kunne fikses, som vi selv havde overset gennem vores kodning, såsom et tomt <h1> element i en af filerne, manglende metatekst, og flere paragraffer der vil være mere semantiske hvis de blev udskiftet med headings.

Disse små ændringer var hurtige at fikse, men er yderst vigtige, da det giver folk med nedsat syns eller andet handicap en bedre oplevelse når de navigere vores kundeserviceside.

## WAVE Web Accessibility Evaluation Tool

- Empty Heading in H1&paragraph-komponent
- Exchange P tag to H3 tag i FAQ-komponent
- Exchange P tag to H3 tag i 2x2-komponent
- Missing Form Label on MyReservation-komponent
- Make stars on Contact a little more red



# Tænke Højt-test

Vi har gennemført endnu to Tænke Højt-test<sup>6</sup> med to af vores tidligere testpersoner, som del af vores testproces for vores webløsning. Formålet ved at teste vores nye webløsning med endnu en Tænke Højt-test, er for at indsamle holdninger og vurderinger af vores redesign af kundeservicesiden. Dette giver os muligheden for at lave små ændringer inden at webløsningen færdiggøres.

Vi valgte at udføre denne Tænke Højt-test på to af vores tidligere testpersoner, og stillede dem de samme spørgsmål som i den tidligere test. Begge testpersoner var positivt anlagt over for vores webløsnings design, og fandt det inviterende og luftigt.

Det betød dog, at den nye Tænke Højt-test ikke bidrog med nye indsigter til mulige forbedringer. Dette kan skyldes en vis partiskhed fra vores testpersoner, fordi de har den tidligere test at sammenligne med samt en potentiel tilbøjelighed til at imødekomme vores løsning.



# Heuristisk Evaluering

En heuristisk evaluering er en usability-metode, som ofte udføres af en UX (User Experience) professional. Det er en effektiv og hurtig måde at afdække de fleste usability-problemer på en webside. Eksperten gennemgår grænsefladen og identificerer problemer ved at vurdere, hvor godt designet overholder anerkendte principper, som kaldes heuristikker. Disse heuristikker kan inkludere Jakob Nielsens 10 heuristikker for brugervenligt design, som omfatter retningslinjer som:

1. Synlighed af systemstatus
2. Match mellem system og den virkelige verden
3. Brugerkontrol og frihed
4. Konsistens og standarder
5. Fejlhåndtering
6. Forebyggelse af fejl
7. Genkendelse frem for erindring
8. Flexibilitet og effektivitet
9. Æstetisk og minimalistisk design
10. Hjælp og dokumentation

Under evalueringen identificerer eksperten problemer ved at gennemgå grænsefladen og notere de steder, hvor designet afviger fra principperne. Hver afvigelse/problem vurderes og prioriteres baseret på hvor alvorligt og betydningsfuldt det er for brugeroplevelsen. Formålet med metoden er at give konkrete anbefalinger til forbedring af websidens brugervenlighed, hvilket i sidste ende kan føre forbedre brugeroplevelsen.

Ved at udføre en heuristisk evaluering kan man hurtigt få en omfattende forståelse af, hvor der er potentiale for forbedringer, og dermed kan siden effektivt optimeres lancering.



Til vores evaluering har vi udvalgt Synlighed af Systemstatus, Konsistens og Standarder, og Æstetisk og Minimalistisk design, som de heuristikker, testerer skal se efter.

I det følgende vil vi give nogle eksempler på de identificerede problemer.

### 1. Der mangler brugerfeedback når der trykkes "claim" på en genstand på hittegodslisten.

Synlighed og systemstatus

Dette er klart en mangel i den nuværende løsning. Til at løse dette i vores konceptuelle løsning, kan der implementeres en modal med info. I en endelig implementering ville knappen føre brugeren til den kontaktformular, som ender hos hittegodsafdelingen.

### 2. Når kontaktformularen indsendes, får brugeren ingen feedback om det er lykkedes.

Synlighed og systemstatus

Dette kan løses ved en pop-up besked eller ved at ændre formularens areal til at blive en feedback besked til brugeren.

### 3. Language-switcher dropdown lukker ikke af sig selv efter der er valgt sprog.

Konsistens og standarder

Dette blev løst ved at tilføje en `setTimeout`-funktion til at håndtere funktionskaldet. Tilsyneladende fordi det pågældende state blev nulstillet undervejs.

```
const switchLanguage = (selectedLang) => {
  setLang(selectedLang);
  setIsOpen(false);
};
```

```
const switchLanguage = (selectedLang) => {
  setLang(selectedLang);
  setTimeout(() => setIsOpen(false), 0);
};
```

### 3. I FAQ-accordion er der ingen synlig forskel når musen føres over de tilhørende links, fordi begge elementer har `{cursor: pointer;}`.

Konsistens og standarder

Dette blev løst ved at give linket en tydelig hover-effekt, nemlig ændring af baggrundsfarven.

Opmærksomheden på dette førte også til en gennemgang og ensretning af alle sitets box-shadows inklusive deres hover-effekt-modstykker.

Den fulde evaluering kan ses i bilag i filbilag.

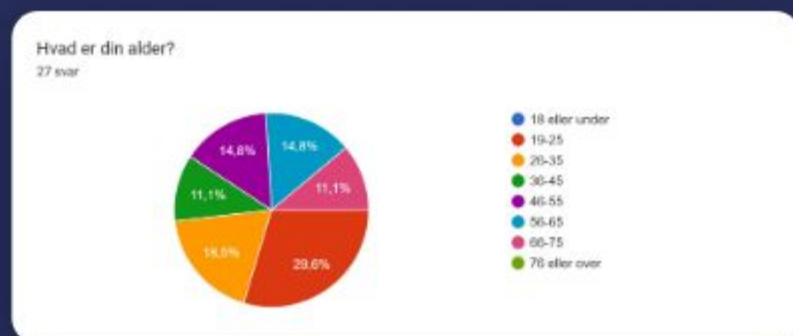


# BERT

For at vurdere vores webløsning bedst muligt, har vi lavet endnu en Bipolar Emotional Response Test, med henblik på at sammenligne resultaterne fra vores tidligere test, og stille dem op mod det mål, vi fik fra vores kvalitative interviews med kunden. Vi genbrugte spørgsmålene fra den tidligere test, for at kunne sammenligne vores resultater direkte med hinanden.

Resultaterne af vores test viser at vi er kommet tættere på at kunne imødekomme vores kundes ønske på alle punkter. Resultaterne indikerer, at vi har formået at skabe en mere minimalistisk hjemmeside, som stadig er informativt og inviterende.

Denne test fik 27 besvarelser, som fordelte sig jævnt over forskellige aldersgrupper.



På graferne til højre ses vores resultater af begge vores test, samt afsenders ønske om placering i spændet.







→ **Refleksion**



# Refleksion

Vi har gennem syv uger arbejdet intensivt på dette projekt med det formål at skabe en omfattende multimedieløsning til CPH Lufthavns kundeserviceundersider. I løbet af projektet har vores indsats været fokuseret på at udvælge, strukturere og præsentere det indhold, der vil give den bedst mulige brugeroplevelse for CPH's kunder. Dette har samtidig indebåret at fjerne indholdselementer, der enten forvirrede brugerne eller unødigt øgede arbejdsbyrden for kundeserviceagenterne.

Vores redesign er ikke kun en opdatering af sidens æstetiske udseende, men en løsning, der balancerer mellem at løse designmæssige udfordringer og efterleve kravene i CPHs visuelle identitet. I vores arbejde har vi forsøgt at skabe en webløsning, som lever op til disse krav, men samtidigt løser de brugermæssige udfordringer, som vi har identificeret undervejs.

Fordi vores løsning er en lille del af en så stor kontekst, ville en endelig implementering være afhængig af samarbejde med flere aktører inden for CPHs organisation. For eksempel vil der være en del arbejde i at forbinde løsningens komponenter med de faktiske databaser og systemer samt at justere dem til disse.

Som beskrevet tidligere i rapporten er der flere features, som vi ikke har nået eller prioriteret at implementere. Derudover er der plads til at lave flere og mere dybdegående brugertests, som vil informere fremtidige iterationers arbejde og optimeringer.

Vi havde planlagt at indhente feedback på vores prototype fra afsenderen og iterere videre på baggrund af den feedback. Desværre har dette ikke været muligt, pga. tidsnød hos begge parter.

Vi vil kunne bruge vores erfaringer fra denne designproces, til at lave en mere struktureret proces i fremtidige projekter. Særligt i overgangene mellem wireframing, mockup og prototyping har været lidt flydende, hvilket har skabt en vis uklarhed i den periode.

På den anden side har dette team på to medlemmer vist sig at være beslutningsdygtigt og vi har kunne bevæge os hurtigt fra idé til beslutning, hvilket har lettet processen.





→ **Litteratur**



# Kildeangivelse

## 1. Design thinking

<https://www.interaction-design.org/literature/article/design-thinking-get-started-with-prototyping>

## 2. Product Benchmark

<https://cmdmethods.nl/cards/library/benchmark-creation>

## 3. Design Patterns

<https://cmdmethods.nl/cards/library/design-pattern-search>

## 4. SWOT

<https://corporatefinanceinstitute.com/resources/management/swot-analysis/>

## 5. TOWS

<https://www.mindtools.com/augstul/the-tows-matrix>

## 6. Tænke Højt-test

<https://www.userdesign.dk/usability-test/taenke-hojt-test/>

## 7. BERT

<https://www.uxforthemasses.com/bert/>

## 8. Google Lighthouse

<https://developer.chrome.com/docs/lighthouse/overview/>

## 9. Value Proposition Canvas

<https://www.strategyzer.com/library/the-value-proposition-canvas>

## 10. Astrid Haugs Content Strategy

<https://astridhaug.dk/contentstrategy>

## 11. Simon Sineks Golden Circle

[https://www.ted.com/talks/simon\\_sinek\\_how\\_great\\_leaders\\_inspire\\_action/c](https://www.ted.com/talks/simon_sinek_how_great_leaders_inspire_action/c)

## 12. Tone of Voice

<https://contentmarketinginstitute.com/articles/define-brand-voice/>

## 13. User Stories

<https://www.visual-paradigm.com/guide/agile-software-development/what-is-user-story/>

## 14. Crazy 8's

<https://designsprintkit.withgoogle.com/methodology/phase3-sketch/crazy-8s>

## 15. Wave Accessibility

<https://wave.webaim.org/>

## 16. Skyfish - billedekartotek

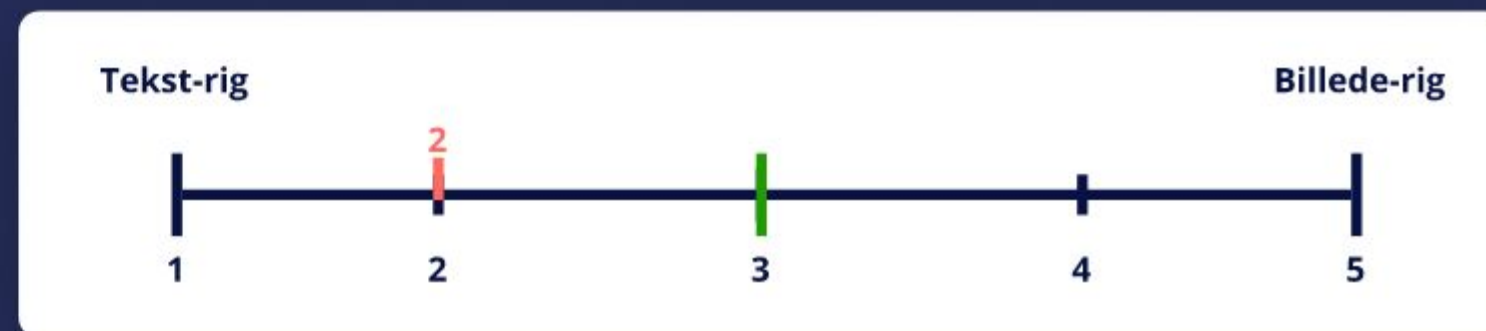
<https://www.skyfish.com/p/cph>







# Bilag 1 - BERT resultater



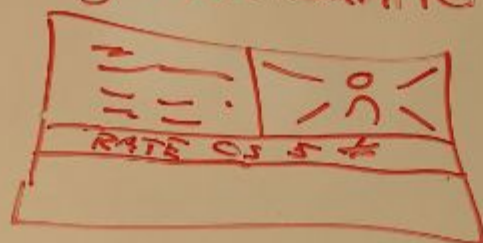


# Bilag 2 - Brainstorm

## Reviewside Banner

- Opfordrer til deling af gode oplevelser på Trustpilot / google reviews
- Kontakt siden [Midt / Bund / Aside]

- Bjørke banner



- Tekst der ændre sig : Forhold til Aktuelle score

"Vi er bedre end 1.7 \* right?"

- Vis gode kommentare



# Bilag 3 - Brainstorm

## Omdirigeringsliste

### Location?

- Midt/Bund af første kundeserviceside
- Spørgsmål & svar

### Design

- Let afkødelig
- Skabe opmærksomhed
- Filtrering?
- Liste > sektion
- teknisk er liste men ligner sektion

### Indhold

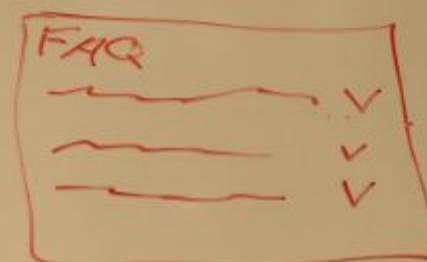
- Telefon numre + Links
- Baggage
- security?
- Pas/Dokumenter
- Polit
- Shopping

## FAQ

Extra Info

- Visninger
- Sidst opdateret

- En samlet Liste der har filtrering/under kategorier eller
- FAQ til hver kategori/underside
- % Billeder, Brug illustrationer



Upvote/Downvote + sortering (Bedst/mest visits øverst)



**Hittegods**

1 Fly

ADCPD

Hvad har du mistet?

iPhone Guld: 15.00

Hittegods Liste

- Top af side
- Mere information
- 72 timer
- Ikoner til hver genstand
- Bedre filtrering
- "Show it Don't say it"
- mere intuitiv
- Kontakt oplysninger til politi Hittegods / Link
- Hvad der bliver beholdt i 4 uger
- Hvad der sendes til Polit: med det samme

72 / Laptop i tasken

tasken  
Gucci: 1.500

24. Jan  
Fundt 13:00 - Fjernet 72  
↓  
48  
↓  
24

Politi Link

Entrepreneurskab og Design



# Bilag 5 - KPI'er

## KPI'er

Antallet af opkald der resulterer i omstilling til anden afdeling, falder med 30%

Trustpilot score skal stige til over 3 stjerner

Antallet af opkald, hvor agenten skal hjælpe med at navigere på hjemmesiden, nedbringes med 80%

Google review score skal stige til over 4.5

Kunder bruger 20% mindre tid på kundeservicesiden (fordi de finder info hurtigere)

Forsinkelse på svartiden for skriftlige henvendelser nedbringes til maksimalt 12 timer (nu 24 timer)

20% flere (nu 40%?) har valgt den korrekte genstand, når de udfylder formular til hittegodslisten.