**Nele Lea Karoline Fyrsten,** Skole ID: nelfyr01

**Emma Møller Klemmensen,** Skole ID: emkle01

Uddannelse:

Multimediedesign

Klasse ID: MMDe23S

3. Semester (2024)

Vejleder:

Malene Patsche

Antal tegn:

**FigJam**: Idégenerering og Konceptudvikling

**Figma**: Design og dokumentation af alle varianter og prototyper

**GitHub**: Dokumentation af kodningsprocessen. Vær venlig og åbne HTML filer med Chrome browseren.

**GitHub Pages**:

**Youtube**: Screenrecording af prototyper og hjemmeside:

Microsoft Teams Planner: Projektstyring, Sprintlog

**Wordpress:** dokumentation.emmaklemmensen.dk/wp-admin/

Username: Malene\_\_Patsche

password: Multimediedesign3eksamen

\*\*Disclaimer\*\*

Vi begyndte at designe på en lille skærm. Så snart vi havde et grunddesign på plads, der passede fint med en lille skærm, skiftede vi til et større skærm og fortog de sidste ændringer på den størrelse. Grunden til at vi skiftede er, at den større skærm er bedre egnet til prototyper.

INDHOLDSFORTEGNELSE

[Design brief 1](#_Toc184384440)

[cerberis 1](#_Toc184384441)

[situation & problem 1](#_Toc184384442)

[project scope 1](#_Toc184384443)

[krav til den digitale løsning 1](#_Toc184384444)

[problemformulering 2](#_Toc184384445)

[projektplan 2](#_Toc184384446)

[produkt 2](#_Toc184384447)

[idéudvikling 3](#_Toc184384448)

[problemløsning 3](#_Toc184384449)

[inspiration 5](#_Toc184384450)

[udforskning: fra variante til prototyper 5](#_Toc184384451)

[brugertests 5](#_Toc184384452)

[kode 5](#_Toc184384453)

[konklusion 6](#_Toc184384454)

[litteraturliste 6](#_Toc184384455)

[bilag 6](#_Toc184384456)

# Design brief

## cerberis

Cerberis er et softwareprodukt, som udvikles af Delfi Tech og maacks IT. Produktet består af en enhed, der kan styre flere varmekilder i et hus. Cerberis er et smart system, som på egen hånd kan skifte mellem aktive varmekilder, tænde og slukke dem, og regulere hver kilde, så alle rum bliver varmet op på til ønsketemperaturen. Det gør det nemt for husejere og viseværter, som styrer flere bygninger, at bruge deres varmekilder mere effektivt og at reducere driftsomkostninger.

## situation & problem

Lige nu består produktet af en enhed, som kan tilsluttes flere varmekilder. Enheden har en skærm, som giver mulighed for at styre varmekilder og se statistiker i forhold til forbrug, priser mm. Indtil videre er denne skærm på selve enheden den eneste mulighed for at styre varmen. Produktet hjælper med at gøre varmestyring i hjemmet mere effektivt, mangler dog fleksibilitet og brugervenlighed, og har nogle begrænsninger i forhold til indstillinger og funktioner på skærmen. Derfor skal der udvikles en app, som kan erstatte skærmen på enheden, og som tilbyder yderligere funktioner.

## project scope

Appen skal være en løsning for alle husejere, uanset om de har en stor teknisk forståelse eller ej, og skal derfor være nemt at bruge og navigere rundt i. Målet er at øge effektiviteten i forhold til varmeforbrug og dermed hjælpe husejere med at spare penge. For at kunne opnå målet kræver appen at have forskellige funktioner, så effektiviteten kan tilpasses individuelt. En af disse funktioner er muligheden for at oprette en tidsplan, som er fokus i dette projekt. Brugeren skal kunne oprette en schedule, så appen automatisk kan styre temperaturerne afhængig af brugerens dagligdag og dermed optimere forbruget.

## krav til den digitale løsning

Appen skal have en brugervenlig og intuitiv brugergrænseflade, der gør det nemt for både teknisk og ikke-teknisk personale at navigere og bruge systemet uden omfattende træning.

Den skal desuden have et dashboard, som kan tilpasses individuelle behov.

## problemformulering

Hvordan udvikles en tidsplan, som er nemt at oprette og redigere i?

Hvordan kan det løses, så brugeren skal beskæftige sig med tidsplanen så lidt som muligt?

Hvordan designes en tidsplan, som er intuitiv at forstå og bruge for teknisk og ikke-teknisk personale?

# projektplan

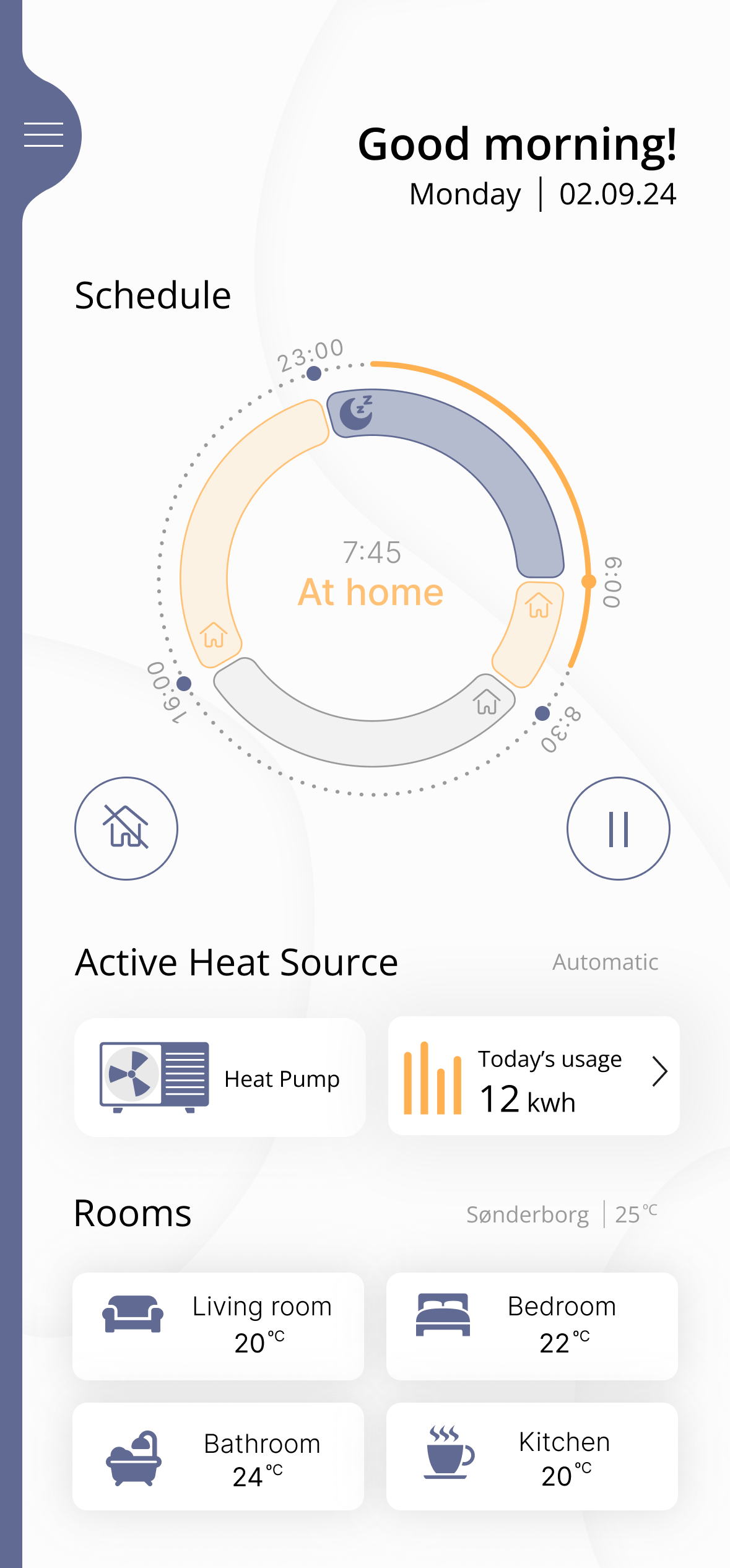
# produkt

## designgrundlag

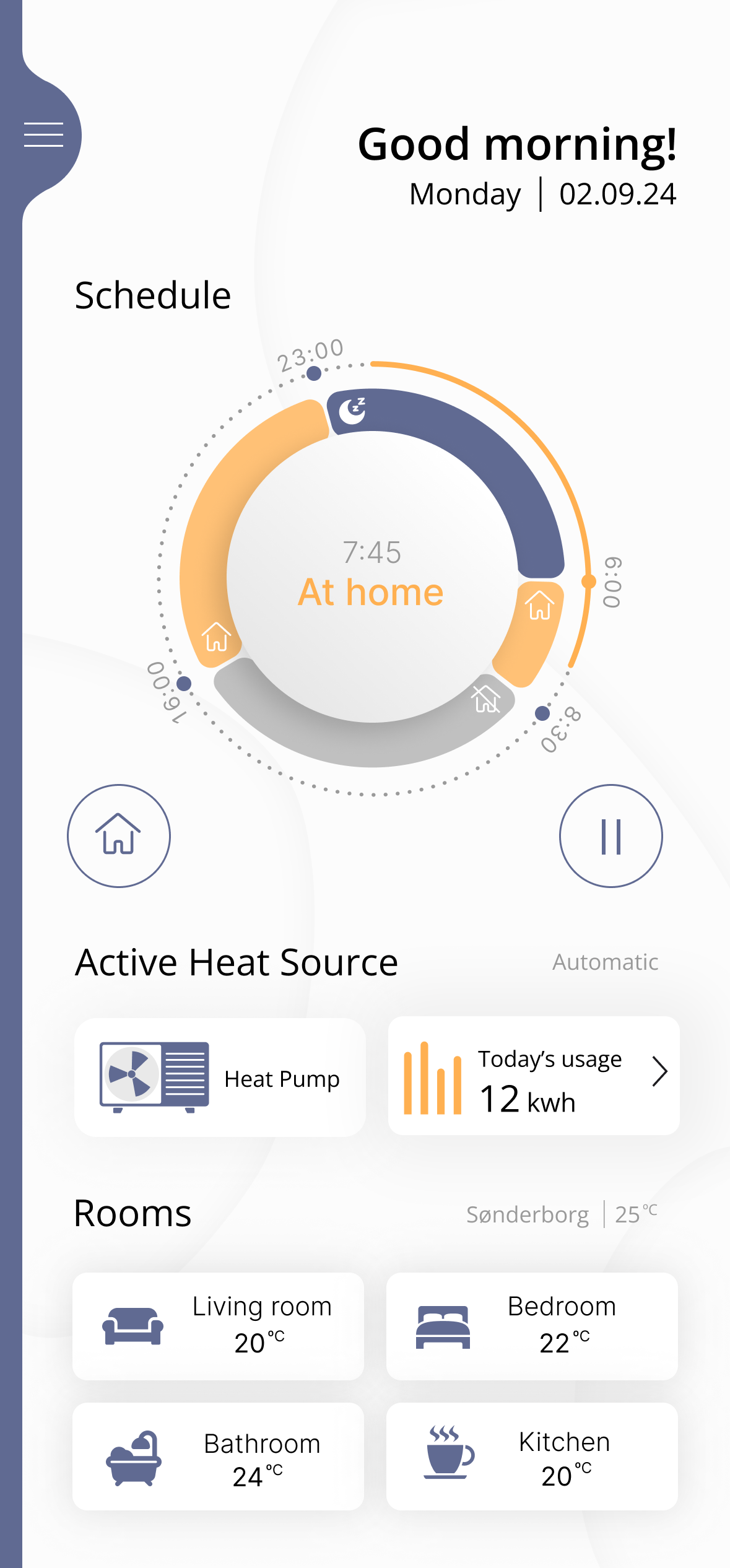
A screenshot of a smart home app

Description automatically generated

Figur 1 - Menu



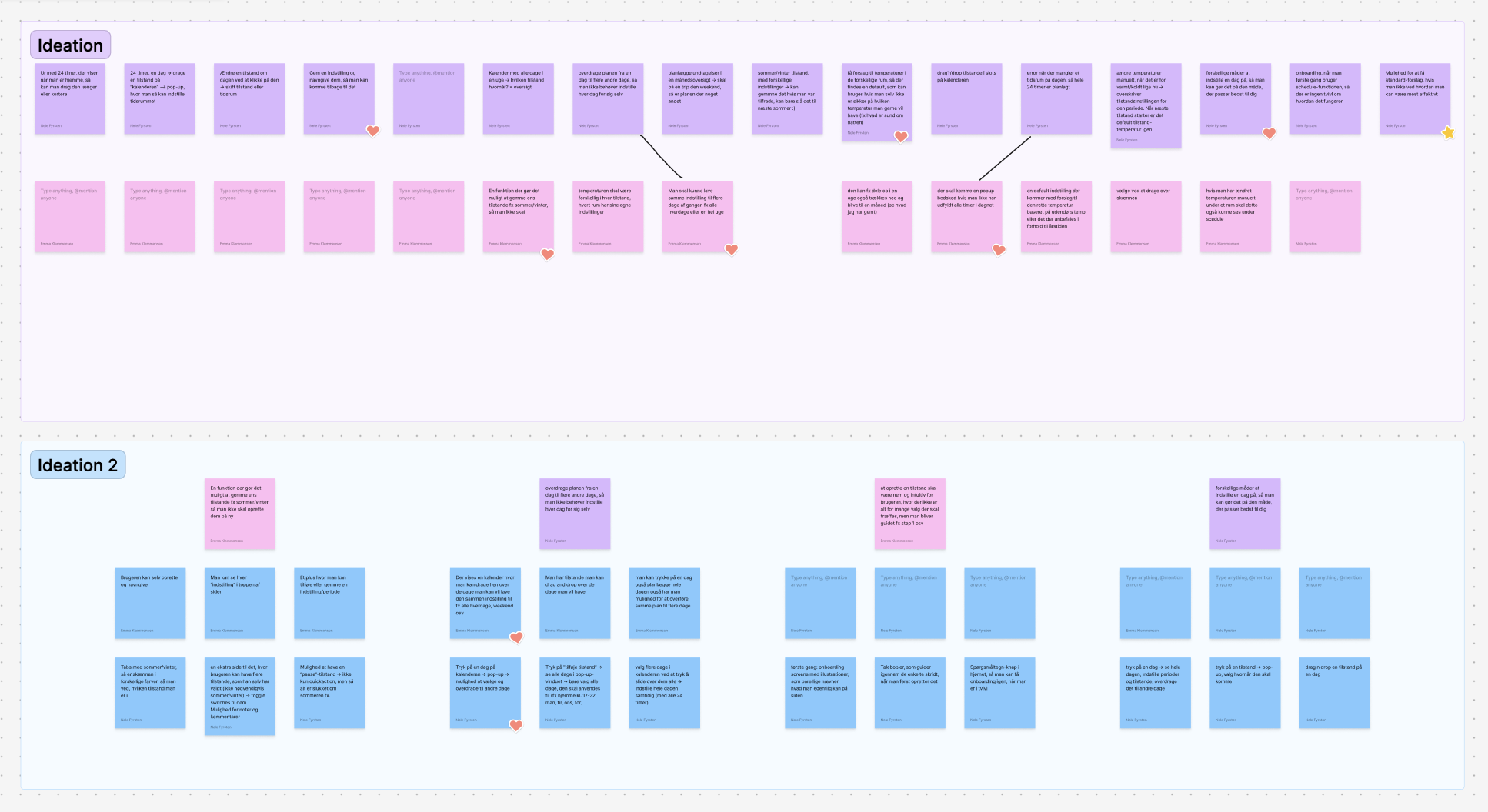
Figur 1 - Oprindelig dashboard



Figur 2 - Dashboard tilpasset

I løbet af virksomhedsprojektet gennem semestret har vi udviklet et dashboard og en menu til appen, som vi tog som udgangspunkt til videreudvikling. Både tilstandene og farverne kunne vi bruge fra denne skærm. For at skabe et genkendeligt design, bruge lignende elementer i hele appen, og dermed øge brugervenligheden, tilpassede vi vores schedule widget på dashboardet (**Figur 2**).

## idéudvikling



Figur 2 - Idégenerering

For vores idégenerering brugte vi ”tøm-hovedet” ideation, hvor vi hver skulle skrive en idé per minut. Eftersom vi allerede havde et dashboard til appen, besluttede vi os for at fokusere på ”schedule”-funktionen. Derfor samlede vi kun idéer til det og ikke til de andre sider og funktioner. Vi prioriterede idéerne og udviklede herefter yderligere idéer på grundlag af det vi havde prioriteret.

Udover det tegnede vi vores idéer på wireframeniveau for at udvikle idéerne endnu mere.

## problemløsning

På basis af idégenereringen udviklede vi et overordnet koncept til schedule siden. Vi videreudviklede dette koncept i løbet af designprocessen, ved at snakke løbende om mulighederne og hvad der fungerede og ikke fungerede.

### Koncept

Funktionens formål er at oprette en tidsplan, der svarer til dagligdagen. Derfor arbejder vi med 3 tilstande:

- ”At home”: når man er hjemme

- ”Asleep”: om natten

- ”Away”: når man er ude af huset

Hver tilstand har deres egne indstillinger i forhold til temperaturen i rummene. Så kan temperaturen i soveværelset fx være mindre om natten. Et andet eksempel kunne være at stuen skal være varm når man er hjemme, man kan godt køles lidt ned, når man er ude af huset, for at mindske forbruget.

Brugeren har en oversigt over dagene og ugerne i form af en kalender, der viser, hvilken tilstand der er aktiv i hvilken tidsperiode. Det kan tilpasses, så ”at home” fx starter kl 15, så rummene er varmere kl 16, når brugeren kommer hjem.

Derudover er det muligt at ændre schedulen til specifikke uger, hvis man fx planlægger at tage på ferie.

Målet er, at brugeren opretter sin schedule en gang i starten, og måske tilpasser den ind i mellem, men meget sjældent. Appen skal styre temperaturer automatisk efter oprettelse, uden at brugeren skal tænke på det.

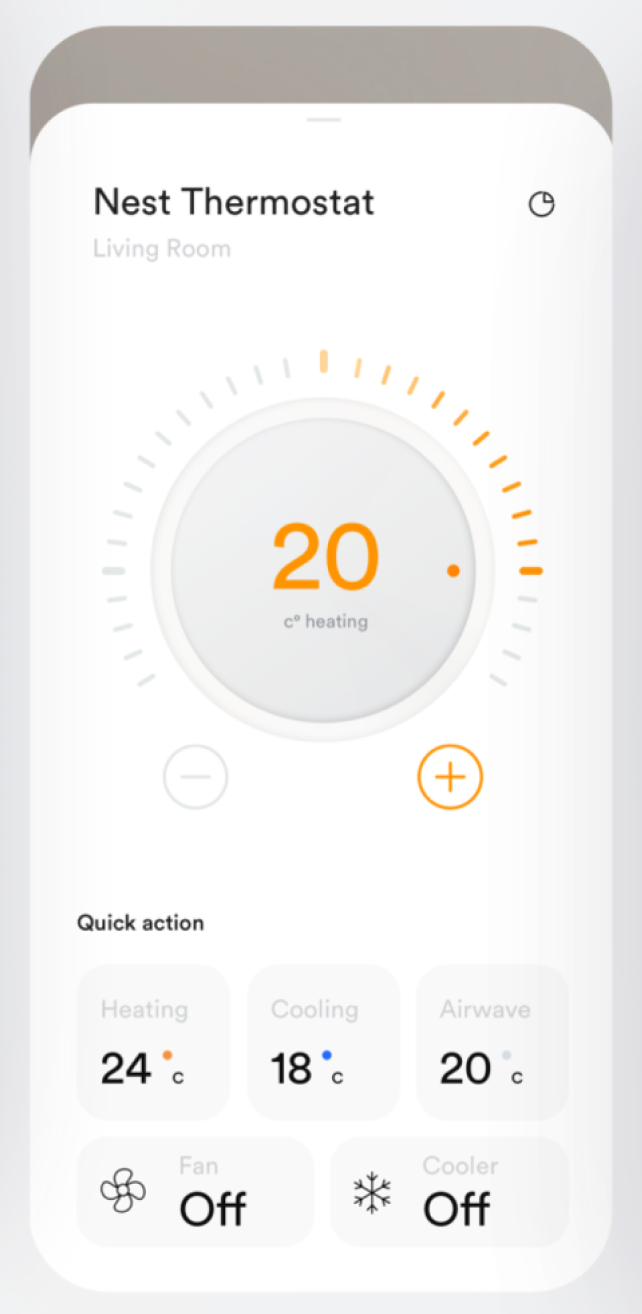
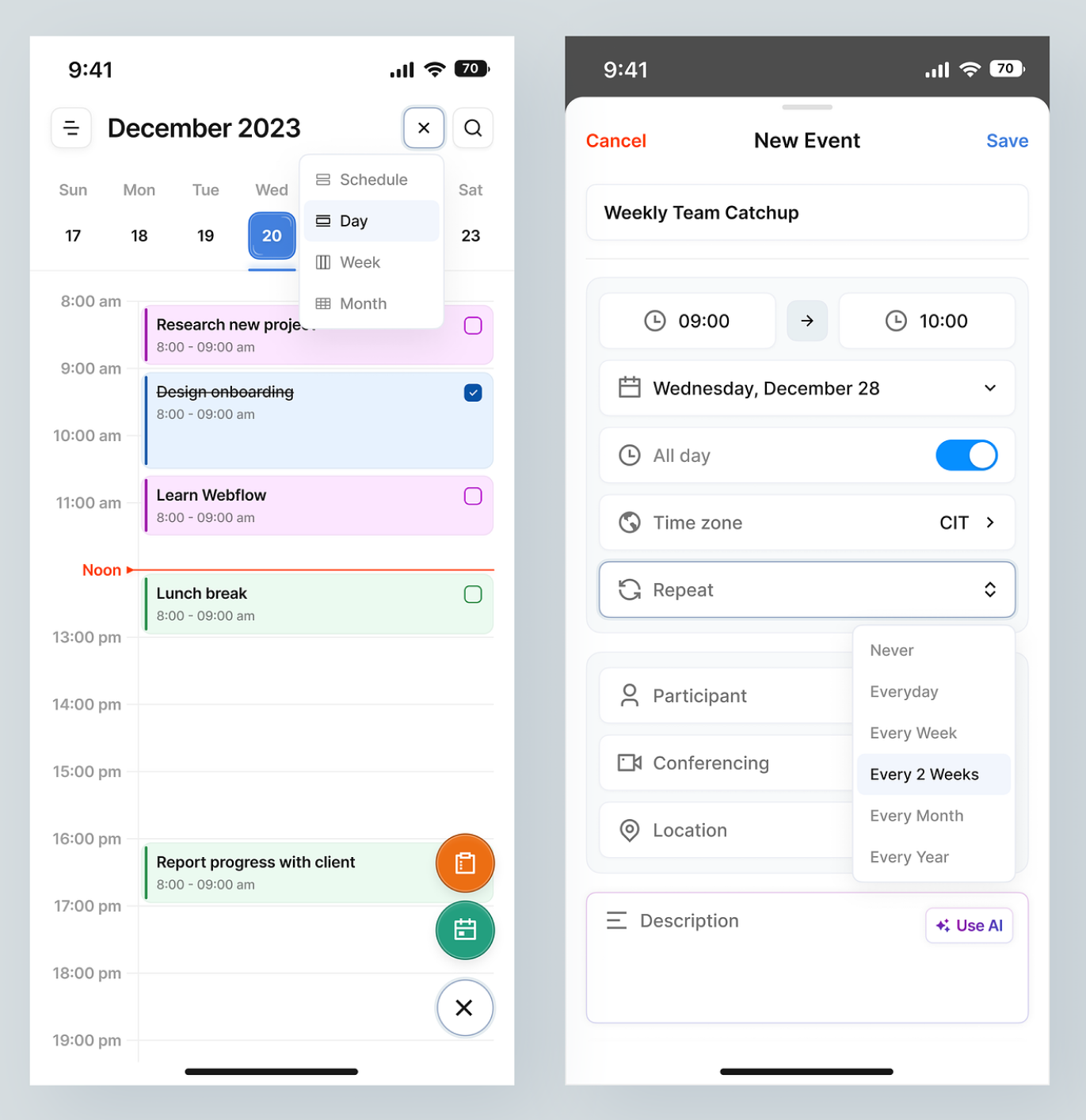
### Ændringer undervejs

Da vi var i gang med at designe, sat vi løbende spørgsmålstegn ved det vi lavede. Det førte til, at vi ændrede konceptet ind imellem.

For eksempel syntes vi i vores oprindelige koncept at brugeren skulle have muligheden for at gemme indstillinger, fx ved at have en ”sommer”- og en ”vinter”-indstilling. Da vi tegnede wireframes, lagde vi dog mærke til at vi ikke alligevel havde brug for det, fordi den optimale temperatur i et rum ikke er afhængig af årstiden og de fleste alligevel slukker for varmen om i sommerhalvåret.

At tegne wireframes på papir skabte i det hele taget meget værdi for vores proces. Vi lagde mærke til at vores overordnede koncept var ret kompliceret, med mange funktioner, som vi ikke rigtig havde brug for alligevel. Derfor reducerede vi funktionerne og opnåede dermed en løsning, der i højere grad mødte målgruppens behov: at være nemt at bruge. Eksamensspørgsmål: hvilke funktioner har vi fravalgt? Hvorfor?

## inspiration



Figur 3 – Inspiration

Vi søgte primært inspiration på dribbble, pinterest og behance. Da hovedfunktionerne er en kalender og temperaturindstilling, søgte vi hovedsageligt inspiration indenfor disse emner.

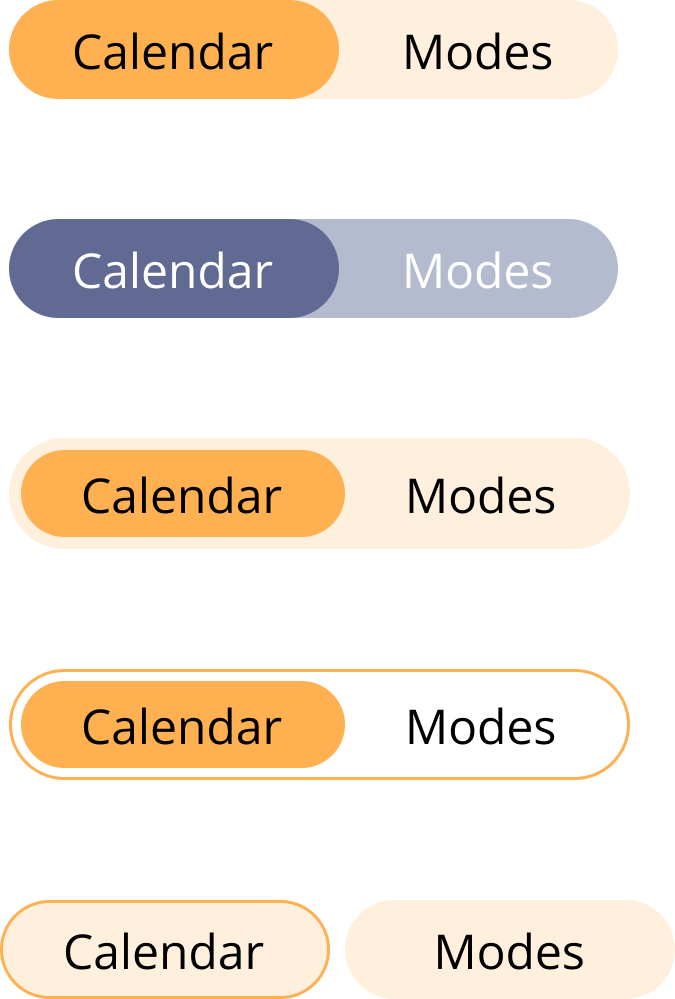
Det, der inspirerede os mest, var et temperaturhjul, som giver temperaturen et visuelt udtryk. Ved at bruge et hjul i stedet for fx kun tallet, øger man interaktiviteten og dermed tilfredshed og brugervenligheden.

I forhold til kalenderen blev vi meget inspireret af måden at vise ugedagene, som kan ses i **Figur 4**. Opbygningen hjælper til et meget struktureret billede og skaber en klar grænse mellem kalenderlabel og aftaler, eller i vores tilfælde, tilstande. I forhold til siden, hvor man kan redigere kalenderen, blev vi inspireret af et design, der arbejder meget med common regions, som gør hele designet meget overskueligt og sikrer, at brugerne ikke bliver frustreret over redigeringen.

## udforskning & design

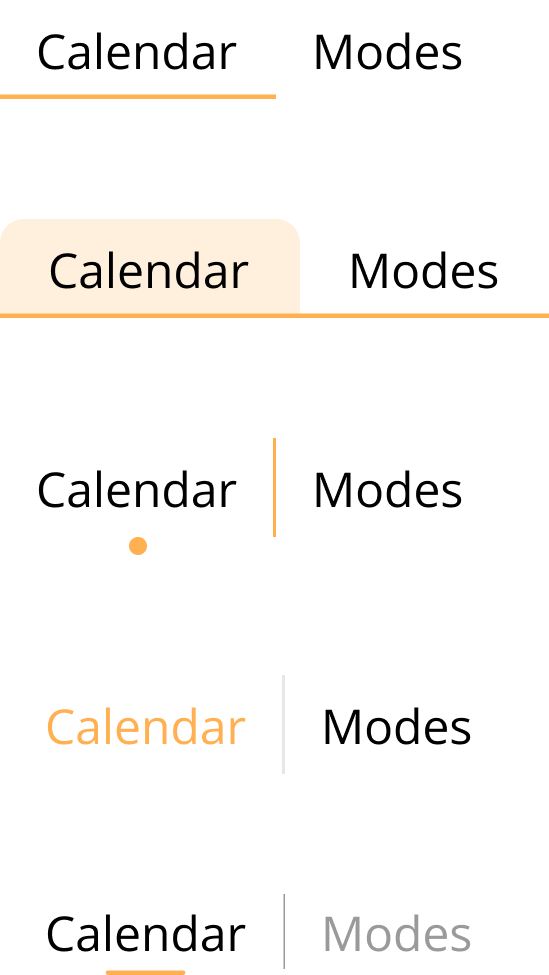
### Tabs

Vi delte schedulesiden i 2 sider: Kalender og Tilstande, Disse blev adskilt ved hjælp af to tabs, hvilket gjorde det muligt at redigere dem uafhængigt af hinanden.



Figur 4 - tabs v1

Figur 5 - tabs v2

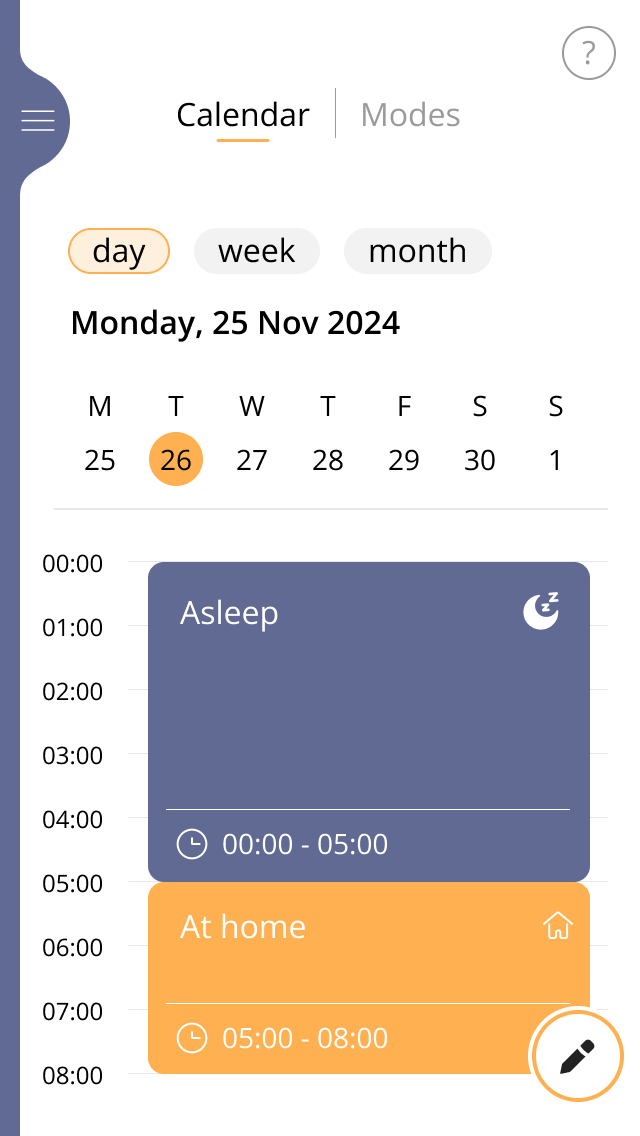


Vi designede to versioner af tabsene. Efter sammenligning af de to versioner fravalgte vi version 1, fordi den ikke passede til den designretning, vi er havde på dashboardet. Den orangegule skal være en accentfarve og er for tungt for en tab, og hele designet fylder for meget på siden. Derfor valgte vi at arbejde videre med den markerede variant. Varianten er meget minimalistisk og let, og distraherer dermed ikke fra sidens andet indhold.

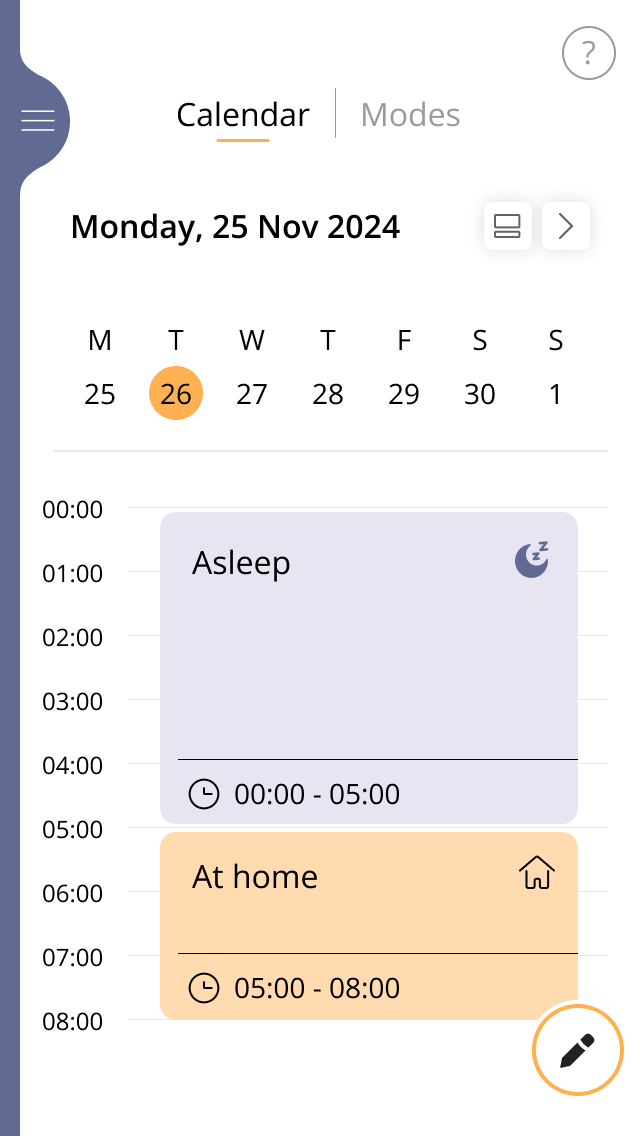
### Kalender

Kalenderen er schedulens ”forside”. Den giver brugeren et hurtigt overblik over deres dagsplan og hvornår hvilken tilstand er aktiv.

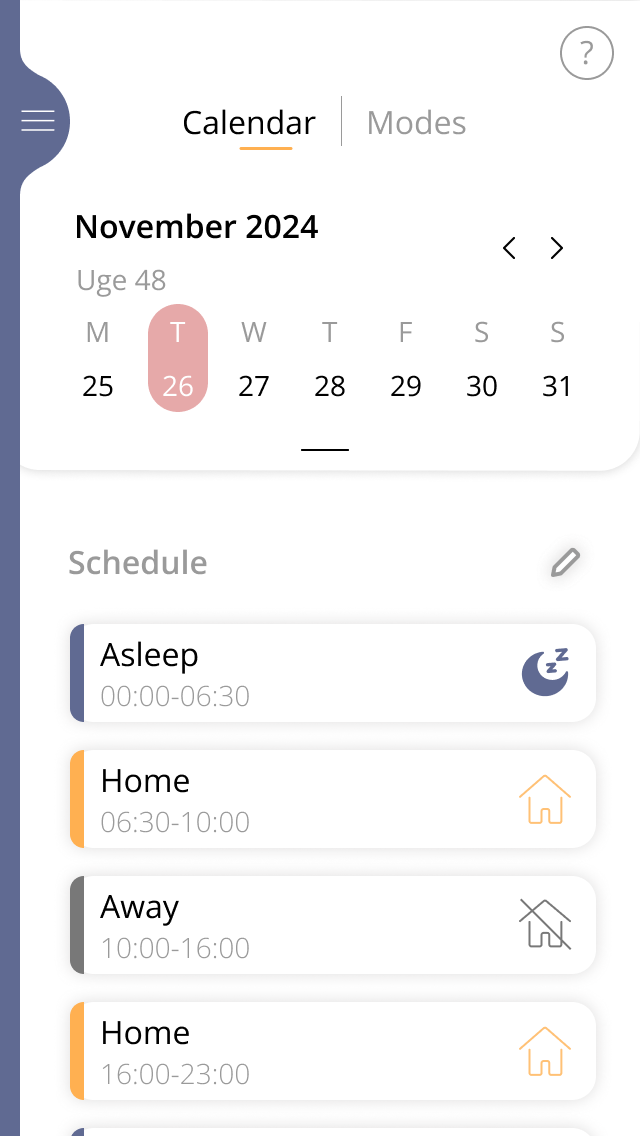
Vores første version var en kalender, der både havde en dags-, en uges- og en månedsvisning. Vi designede flere varianter, der var forskellige både i hvordan tilstandene vises, og hvordan man skifter fra fx dag til måned.



Figur 8 - Kalender v1.1



Figur 8 - Kalender v1.2



Figur 8 - Kalender v1.4

Skift visningen med tabs. Meget stærke farver, som fylder lidt for meget på siden.

Gå en uge frem med pilen og skift til ugentlig eller månedlig visning på ikonet. Ikke intuitivt nok at ikonet har den funktion.

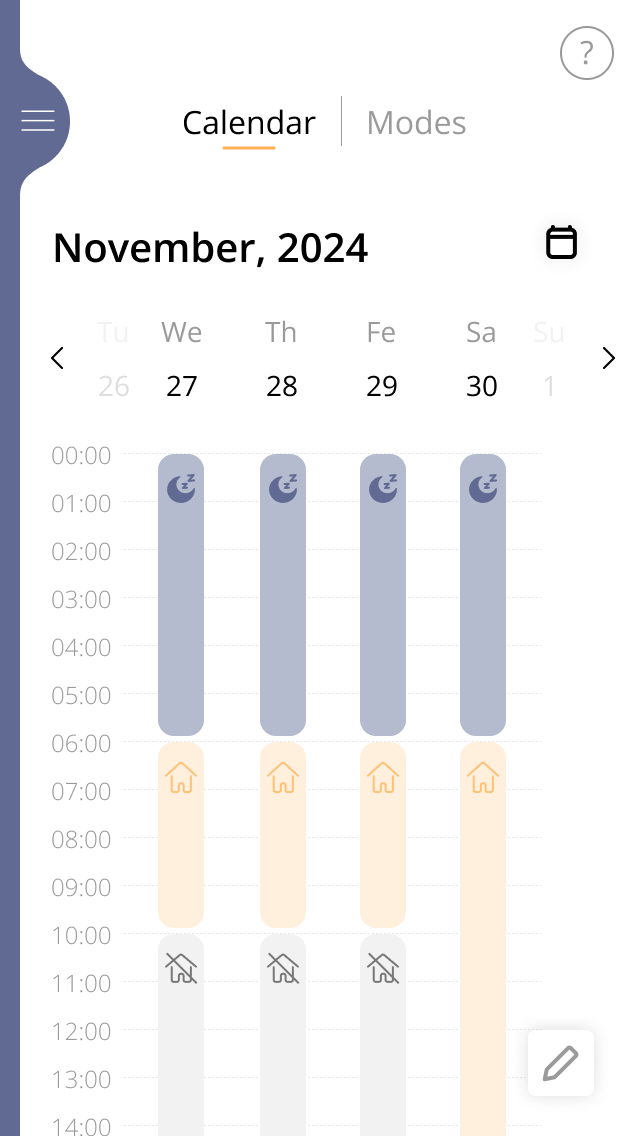
Træk dagene ned for at se hele måneden og vælg en anden dag derfra

I den første version kunne brugeren se tidsplanen for en dag og for en uge. På den månedlige visning kunne man vælge en dag i fremtiden for fx at ændre den. Vi designede ikke den månedlige visning i denne version, fordi vi reevaluerede konceptet. Tilstandskasser fylder hele skærmen, hvilket ikke er nødvendigt, fordi der ikke er særlig mange informationer i en kasse. Desuden er det vigtigere for brugeren at kunne sammenligne alle ugens dage, hvis man fx har et fuldtidsjob, hvor alle hverdage følger den samme tidsplan.

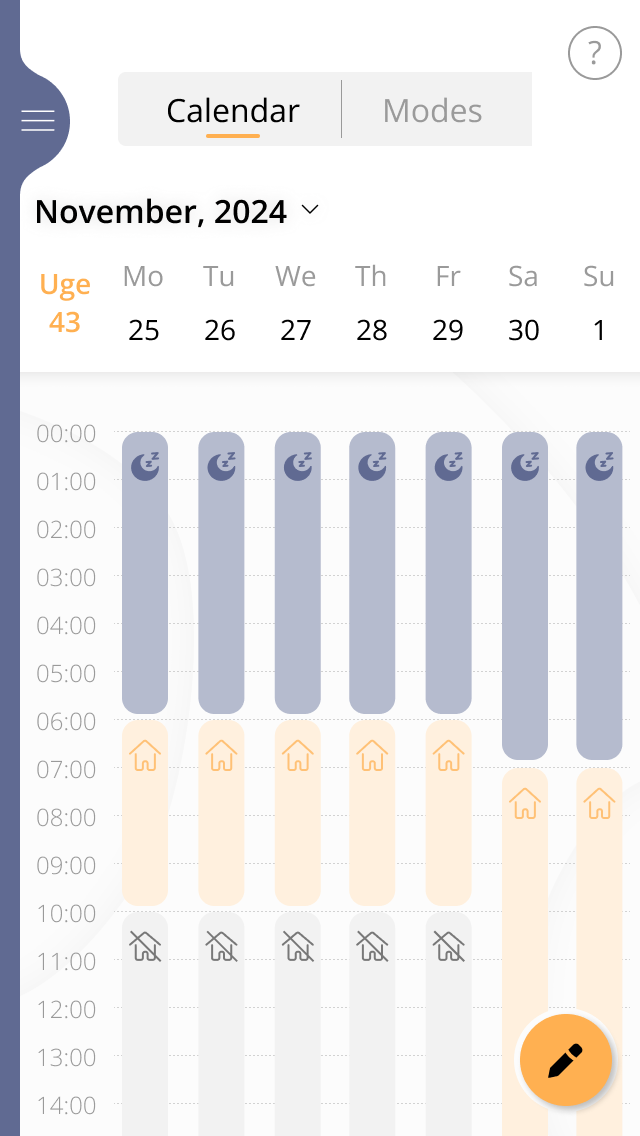
De eneste relevante informationer er hvilken tilstand det er, samt tidsrummet. Det ledte til opdagelsen, at kun at vise en dag, ikke skaber meget værdi for brugeren. Derfor gik vi i gang med at designe version 2.

I version 2 skruede vi ned for funktionerne og fokuserede på visning af en uge, som er nemmere at overskue og bearbejde. For at skabe genkendelighed for brugeren brugte vi de samme farver og ikoner som på dashboardet til de definere de forskellige tilstande. Ved at vise så få informationer om en tilstand, som muligt, kan vi vise meget mere information på hele siden.

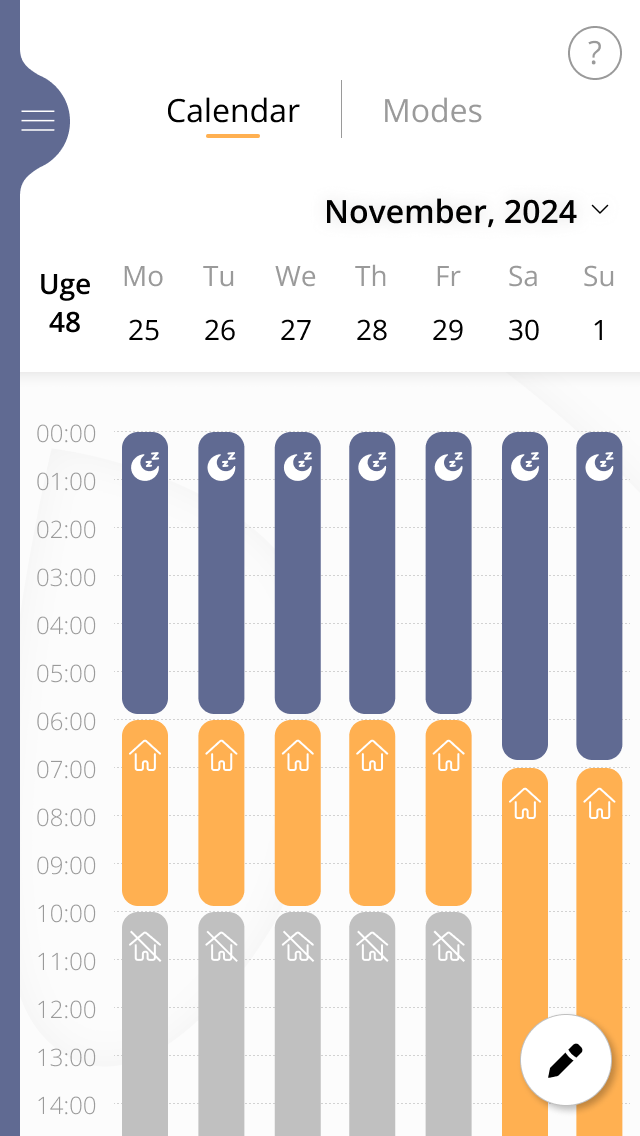
Derudover eksperimenterede vi med forskellige versioner af tabs for at finde den løsning, der fungerede bedst i sammenhæng med kalenderen.



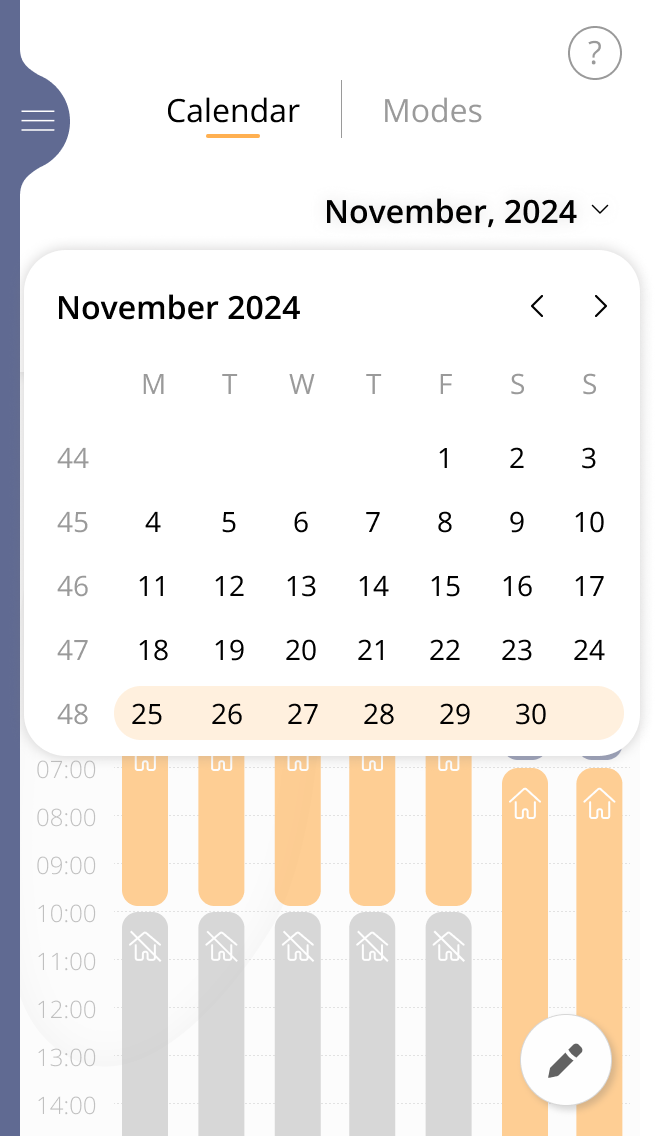
Figur 9 - Kalender v2.1



Figur 10 - Kalender v2.2



Figur 11 - Kalender v2.3



Figur 12 - Kalender v2.3 - måned

For den endelige version af kalenderudforskningen brugte vi kraftigere shades af vores tilstands farver. Det øger kontrasten og gør kalenderen endnu mere nem at overskue. Idet det er en oversigtsside, hvor brugeren går ind enten for at ændre tidsplanen eller bare lige tjekke den, er det afgørende at det ikke tager lang tid at skabe et overblik.

Vi inkluderede desuden muligheden for en månedsoversigt og dermed for at skifte ugen hurtigt.

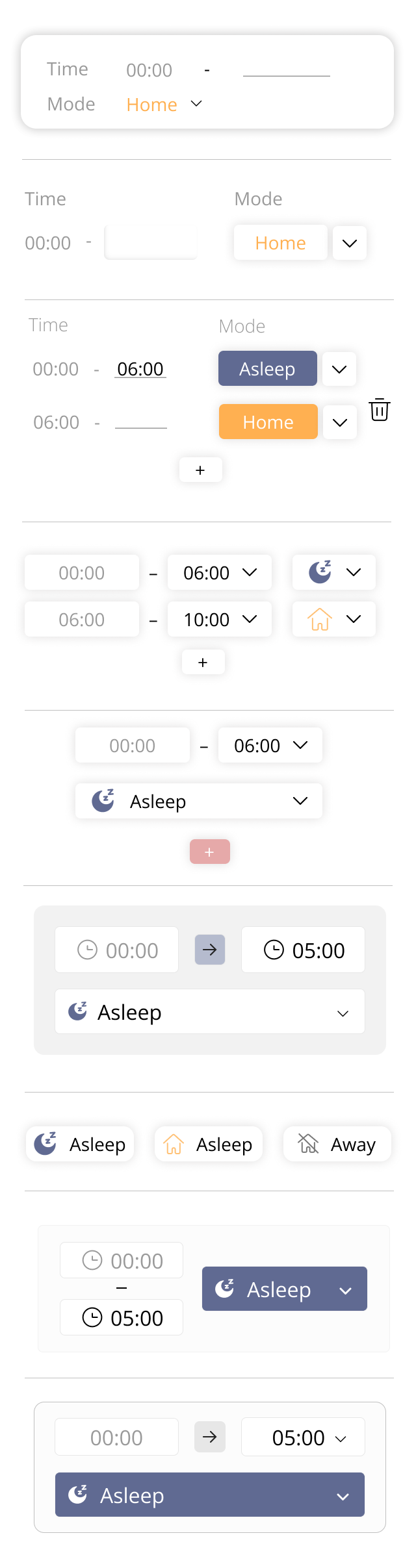
### Redigering

På kalenderen ligger en knap, der leder videre til redigeringssiden. Formålet af denne side er at redigere inde i kalenderen. Det vil sige at man kan oprette nye tidsperioder og tildele nye tilstande til perioderne. På den måde er kalenderen meget fleksibelt og kan tilpasses brugerens behov.

Siden skulle have flere funktioner:

**1. Valg flere dage**

Brugeren skal have mulighed for at vælge flere dage, der skal have den samme dagsplan og indstillinger. Hvis man fx arbejder fuldtid og alle hverdage ser ens ud, kan man indstille det på en gang i stedet for at skulle gøre det enkeltvis.



Figur 13 - Komponent "edit period"

**2. Valg en uge**

For at kunne planlægge en dag i fremtiden, der er forskellige for brugerens standard-indstillinger, kan man vælge en eller flere uger. Hvis man fx har planlagt en weekendtur, så kan man ændre schedulen kun for den periode. Et andet eksempel kunne være, at man skal på ferie i to uger og derfor skal have en anden plan til fx uge 50 og 51, uden at påvirke alle andre uger.

**3. Valg tidsperiode på dagen og tildel tilstand**

Brugeren skal kunne oprette en meget individuel tidsperiode og tildel en tilstand. Det kunne fx være ”kl 8 – kl 16 🡪 tilstand ’Away’”.



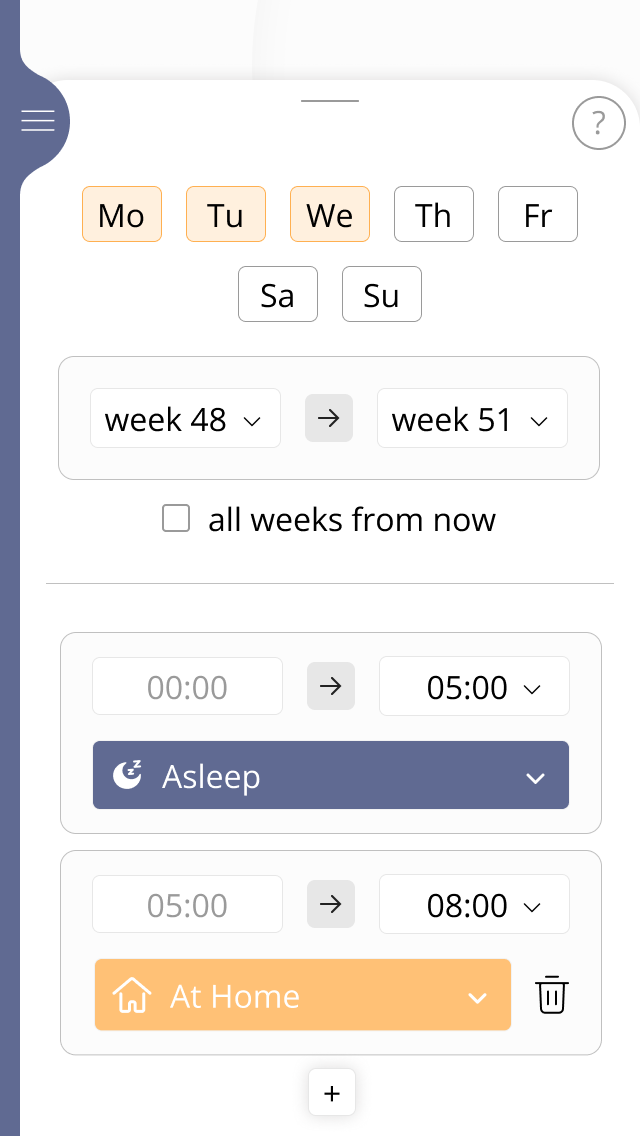
Figur 14 - Komponent "selectDays"

For at finde den bedste og mest intuitive løsning, designede vi mange versioner og varianter af de forskellige komponenter, som kan ses i **Figur 13** og **Figur 14**. Disse versioner sat vi så sammen i skærme, for at se, hvordan det virker på siden og hvilke komponenter fungerede bedst sammen.

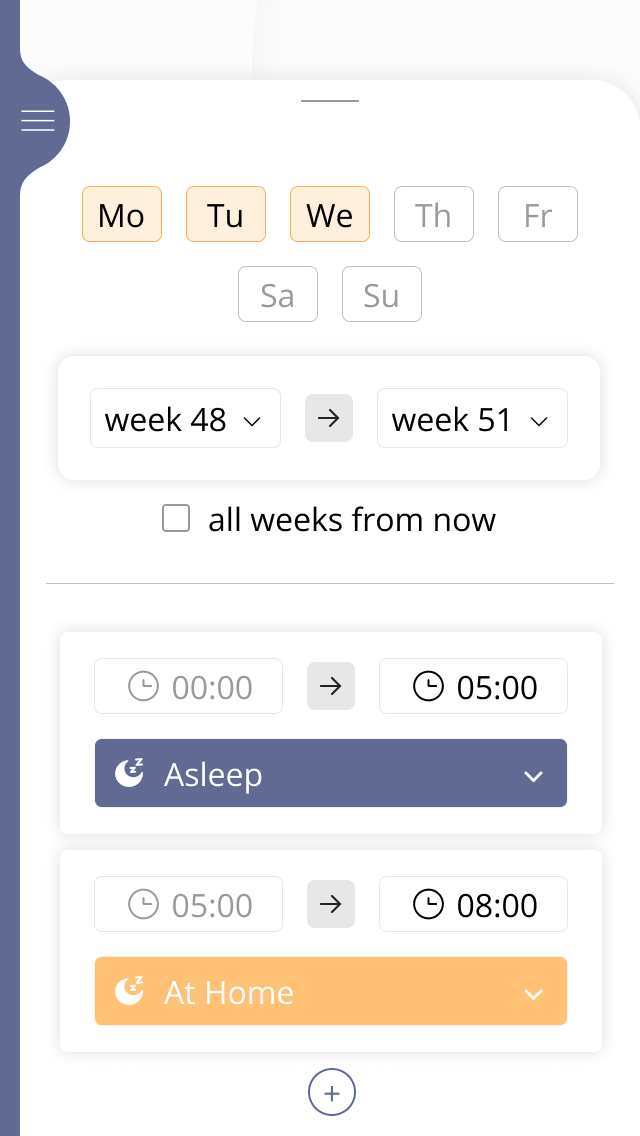
**Figur 15** viser en første version af hele siden. Siden er en overlay, der flyver ind nedefra, og som kan lukkes ved at trække ned igen. Vi fravalgte denne version, fordi dagene er for tæt på hinanden. Det giver både en indsnævret udseende og det er svært at træffe den dag, man ønsker. Det der fungerer godt i denne version, er common regions til tilstandene. Ved at have både tidsperioden og tilstanden i en container, bliver det meget tydeligt, at de to indstillinger hører sammen.

**Figur 16** viser en bedre variant af dagene. Her er det delt i hverdage og weekend, og der er nok plads til hver dag. Tilstandsindstillingen fylder meget mindre her, men er også sværere at overskue en i v1, da hverken proximity, eller common regions bliver brugt her.

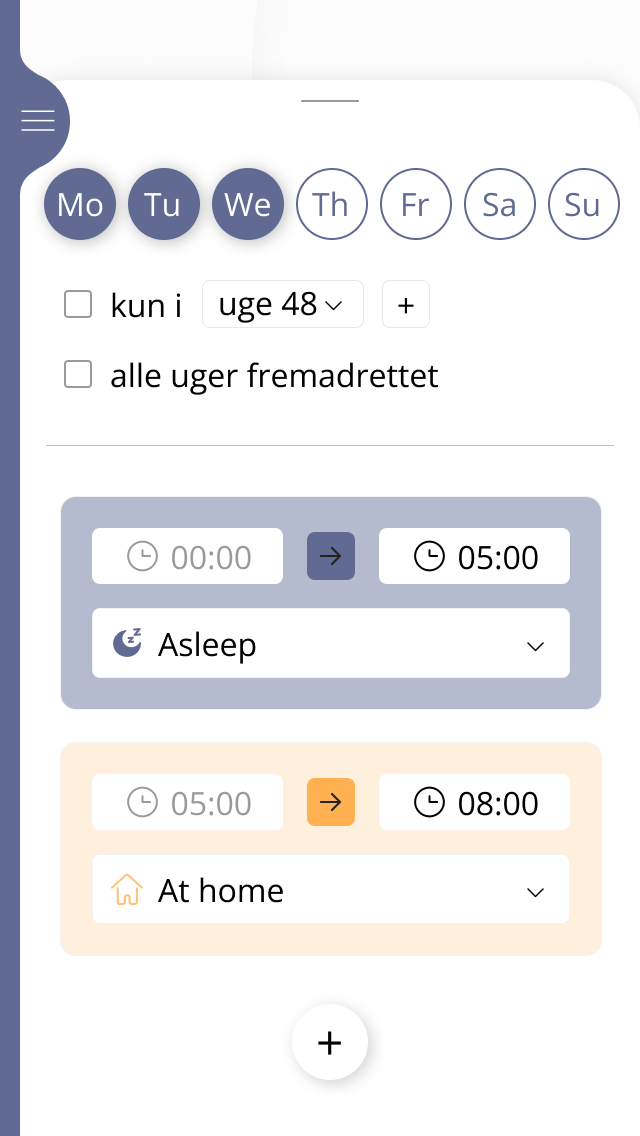
I forhold til at vælge en eller flere uger, støder man hurtig på begrænsninger, fordi det bliver besværligt at tilføje mere end to uger.



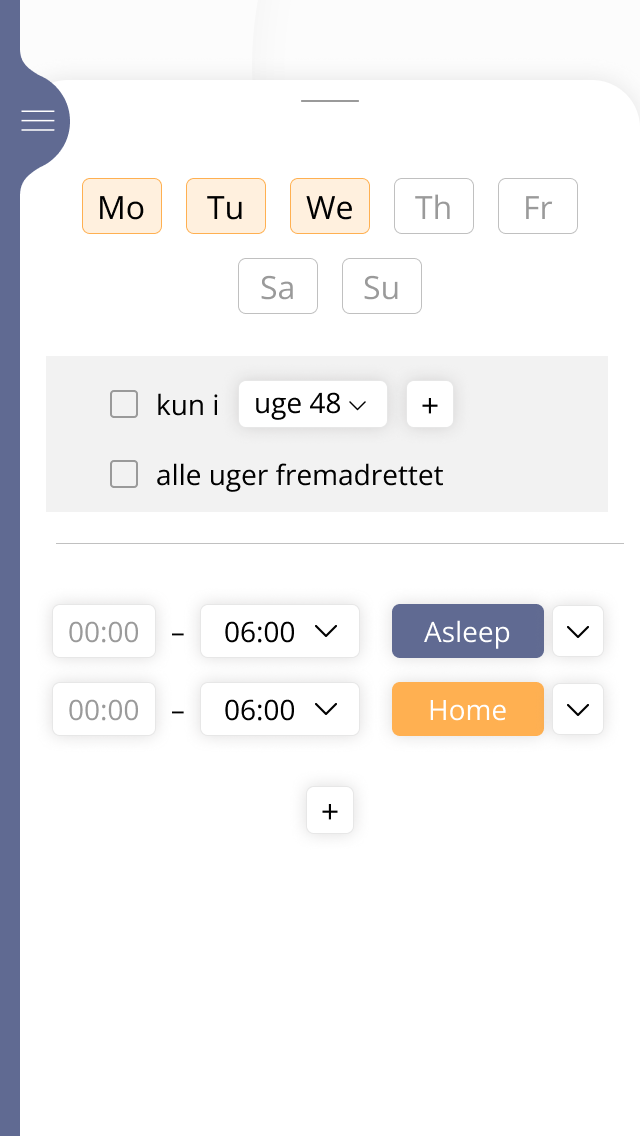
Figur 18 - Rediger v3



Figur 17 - Redigere v3.1



Figur 15 - Rediger v1



Figur 16 - Rediger v2

I version 3 (**Figur 18**) arbejdede vi meget med common regions, for at gør processen så overskuelig som muligt. På den måde ved brugeren, hvilke indstillinger hører sammen til en kategori og har nemmere ved at redigere schedulen.

Derudover udviklede vi et mønster til alle knapper, der har et ”selected”- og et ”not selected”-state, som f.eks. dagene. Sådan en valgmulighed er kendetegnet ved at have en border rundt omkring, og fyldes gul, når den er valgt. Det står i kontrast med **Figur 17**, som bruger container-mønsteret med skygge, som vi også bruger på dashboardet. Vi valgte at skille de to containere ad, fordi den på dashboardet kun er en information, hvorimod her kan der fortages en indstilling.

Screens screenshot of a phone

Description automatically generated

Figur 19 - Redigere endelig version

I forbindelse med vores tanker om brugerrejsen (se side blabla) forstod vi, at der er rigtig mange ting på siden, som skal forstås samtidige. Derfor anvendte vi både Hick’s lov[[1]](#footnote-2) og Miller’s lov[[2]](#footnote-3), for at skabe en mere tidfredsstillende oplevelse for brugeren.

Vi anvender chunking ved at bruge indstillinger til dagene og ændringsperioden som første trin, og indstillinger til tilstandene på selve dagen, som andet trin. Dermed mindsker vi beslutningerne på en side samt den kognitive belastning.

For at gøre brugerrejsen endnu nemmere, tilføjede vi overskrifter og korte beskrivelser, så man forstår hvad indstillingen går ud på.

I forhold til at indstille en ny tilstand, så starter indstillingen altid kl 00:00. Så snart man indstiller sluttidspunktet og tilføjer en ny tilstand/periode, synkroniseres starttidspunktet med det sidste sluttidspunkt. Starttidspunktet kan altså aldrig ændres i selve perioden og har derfor en grå farve. På den måde sikrer vi, at der ikke opstår huller i dagsplanen. Eftersom vi har 3 klar definerede tilstande, er det vigtigt at hele dagen er planlagt.

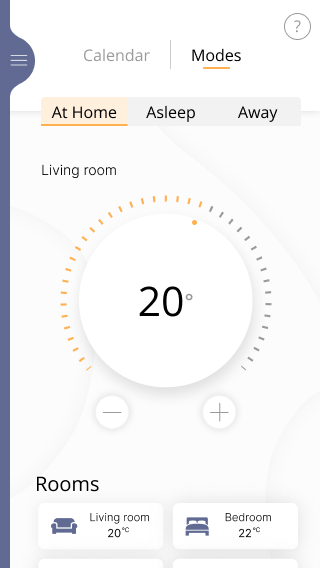
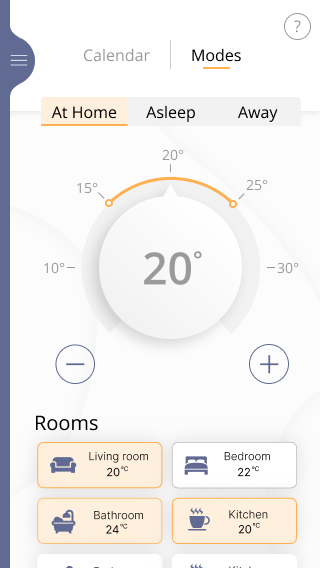
### Tilstande

Som nævnt tidligere blev vi meget inspireret af et temperaturhjul og ønskede derfor at integrere denne funktion, når brugeren skal indstille temperaturen i de forskellige rum til de forskellige tilstande. I de første versioner udforskede vi forskellige muligheder for, hvordan denne funktion kunne se ud, hvilket fremgår af eksemplerne herunder.

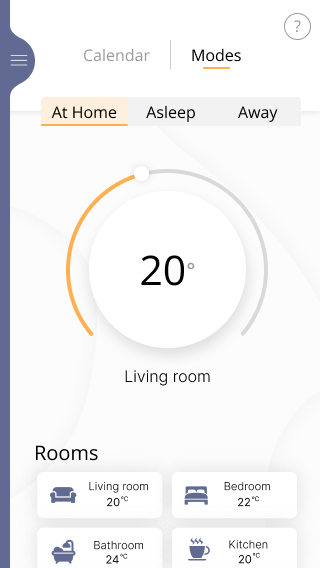
Figur 13 – tilstand v1.1

Figur 15 - tilstand v1.3

Figur 14 – tilstand v1.2



Figur 13



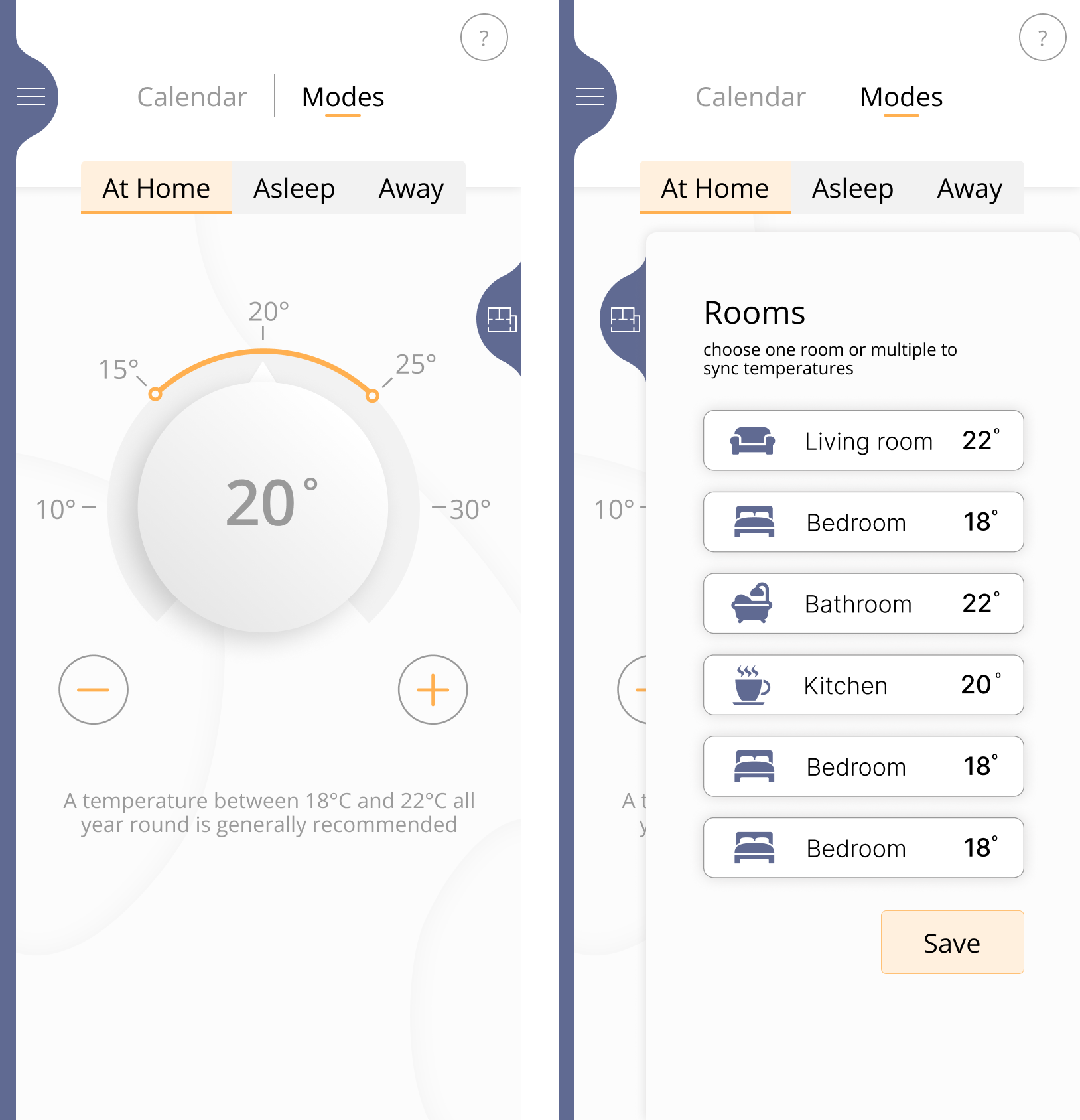
De første versioner af temperaturhjulet blev udviklet med et minimalistisk design, som ses i **Figur 13** og **Figur 14,** for at skabe en visuel sammenhæng med resten af vores design.

I tredje version som ses i **figur 15** har vi tilføjet flere detaljer, herunder temperaturangivelser og en orange linje, der markerer de anbefalede minimums- og maksimumtemperaturer. Disse justeringer blev lavet med vores sekundære målgruppe i tankerne, da vi ønskede at give temperaturhjulet et mere teknisk og funktionelt udtryk.

Over temperaturhjulet har vi placeret tre tabs, der repræsenterer de forskellige tilstande. Dette gør det muligt for brugerne hurtigt at skifte mellem tilstandene, samtidig med at det tydeligt fremgår, hvilken man er ved at redigere. Herefter kan brugerne indstille de ønskede temperaturer for hvert rum.

Vi har genbrugt rumkomponenterne fra vores dashboard, men har tilføjet vores mønster med border som indikere, at brugeren kan vælge elemnet og en selected state, når et element er valgt. Dette gør det muligt for brugerne at vælge flere rum og indstille den samme temperatur for flere rum.

Efter en dialog og en ekspert test med vores underviser, Malene, om tilstandenes funktionalitet, blev vi opmærksomme på, at brugerrejsen kunne forbedres. Det blev klart, at det ikke er tilstrækkeligt intuitivt for brugerne, at de først skal vælge et eller flere rum og derefter indstille temperaturen. Derfor eksperimenterede vi videre i forhold til hvordan vi kunne simplificere brugerrejsen.



Figur 16 - version 2.1



Figur 17 - version 2.2

Som **Figur 16** illustrerer, eksperimenterede vi med at designe en aside til rummene, men dette blev senere forkastet, da det ikke opfyldte vores mål om at gøre brugerrejsen mere simpel og intuitiv. I stedet ændrede vi på indholdets rækkefølge ved at placere rummene før temperaturhjulet, som vist i **Figur 17**, for bedre at matche trinnene i brugerrejsen.

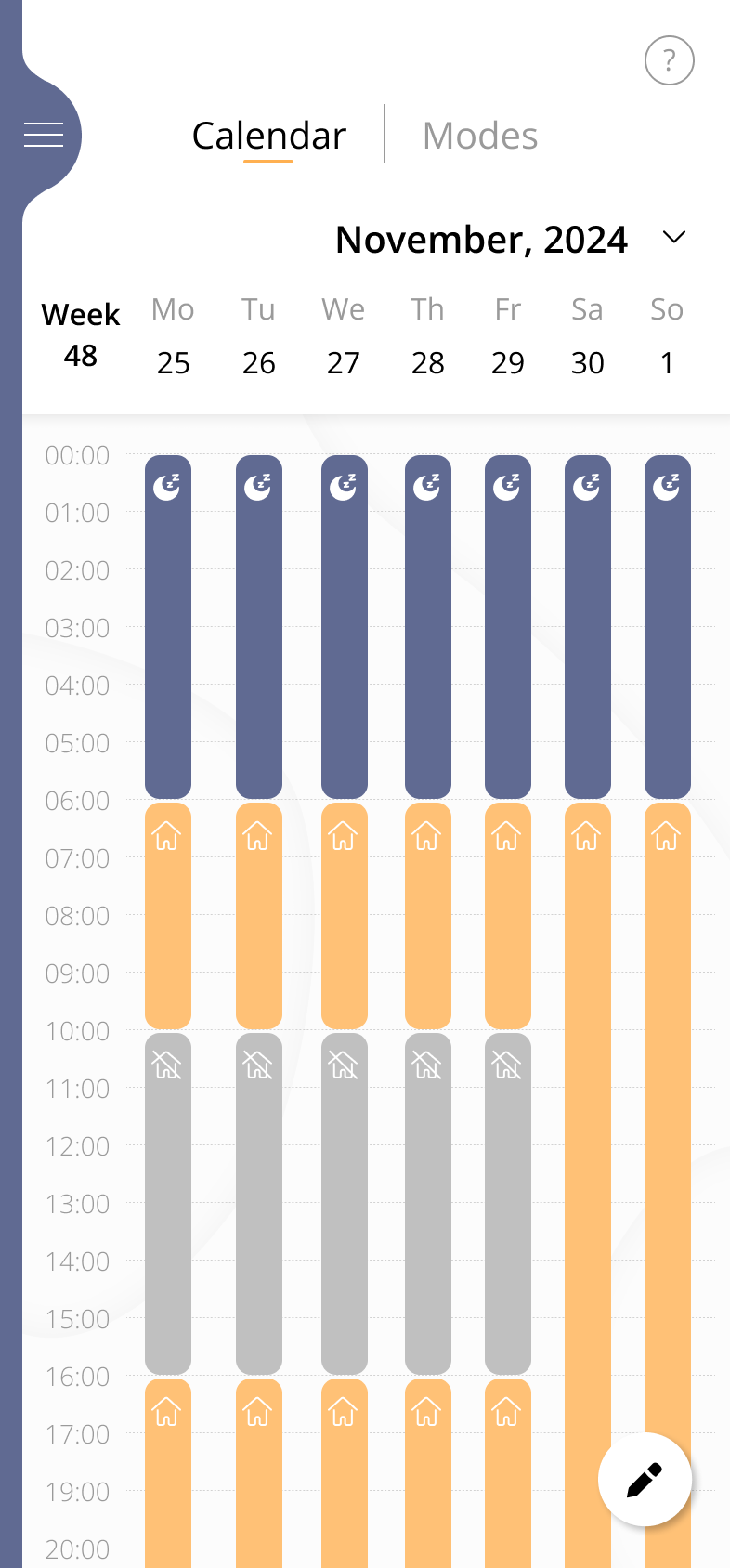
Vi opdagede dog, at denne tilgang resulterede i en overfyldt side med for meget information på én gang. Dette førte os til at anvende Hick's lov (Laws of UX, 2024). For at reducere kognitiv belastning og minimere antallet af valg, delte vi siden op i flere trin og fordelt indholdet over flere sider. Denne opdeling gjorde det lettere for brugeren at navigere og fokusere på ét trin ad gangen.



Figur 18 - version 3.1

I version 3, som er den endelige version, har vi ændret rumkomponenterne, som vist i Figur 18, så de fylder mindre vertikalt. Vi har også tilføjet en checkbox i stedet for at bruge en ”selected”-state for rumelementet. I den her sammenhæng gør det det mere tydeligt, at brugeren kan vælge rummene. Dette designvalg blev truffet for at skabe genkendelighed for brugeren i tråd med Jacob's lov.

## brugertests



Figur - brugertest af schedule

løbet af hvert sprint i designprocessen har vi løbende udført mindre tests. Disse tests har både omfattet evaluering af enkelte komponenter og afprøvning af ideer til det overordnede koncept. Blandt andet har vi gennemført en Standalone Design Critique (NNGroup, 2018) internt i gruppen, hvor vi diskuterede konceptet og vurderede, hvilke funktioner der var nødvendige. Hele processen har været fokuseret på at holde den primære bruger i centrum.

For at teste konceptet bag schedulesiden udførte vi en *Think Aloud*-test for at undersøge, om brugerne forstod sidens formål og kunne aflæse dens indhold korrekt. Overordnet viste resultaterne[[3]](#footnote-4), at testpersonerne forstod schedulesiden og de funktioner, vi havde implementeret.

Det er dog vigtigt at være kildekritiske i forhold til testen. Vores spørgsmål var til tider ret ledende, hvilket kan have påvirket brugernes svar og forståelse under testen.

Dette har skabt stor værdi for vores designproces, d

# Endelig design

På baggrund af det vi har fået ud af brugertestene, satte vi spørgsmålstegn ved vores løsning og videreudviklede vores koncept.

I den oprindelige version var der to steder, hvor man kunne redigere tingene: dagsplanen på redigeringsknappen i kalendertab, og temperaturen i modes-tab. Fordi testpersonerne var meget i tvivl om, hvor de skulle indstille de forskellige ting, besluttede vi os for kun at have et sted, hvor man kan fortage redigeringer og indstillinger.

Derfor fjernede vi tabs og redesignede redigeringsknappen på forsiden, så den var mere tydeligt.

A screenshot of a graph

Description automatically generatedEt billede, der indeholder tekst, skærmbillede, Font/skrifttype, design

Automatisk genereret beskrivelse

# kode

# konklusion

# litteraturliste

<https://lawsofux.com/hicks-law/>

# bilag

Wireframes tegninger

1. <https://lawsofux.com/hicks-law/> [↑](#footnote-ref-2)
2. <https://lawsofux.com/millers-law/> [↑](#footnote-ref-3)
3. Resultat brugertest: https://www.figma.com/design/GXJw5Q5h6drflFfEr0S2Mw/Nele-%26-Emma?node-id=1175-16772&node-type=canvas&t=Qcllr8vCdwvtgMdZ-0 [↑](#footnote-ref-4)