
Projekt 1 - Krisen kradser

Efterårssemester 2024

Jeppe Bøgeskov Bech
jepp9920@zbc.dk

Alexander Schade Knudsen
alex245h@zbc.dk

Andreas Jensen
andr328q@zbc.dk



**Vær med.
Verden er til at forandre.**

2. x

ZBC Handels- og Teknisk gymnasium Slagelse
Akademisk år 2024-2025
7. december 2024

1 Abstract

This report encompasses the project of developing an application that can be used to manage crises. The application will be developed by a new company called "KEP". The application will be developed using a modern application development solution, ensuring optimal performance and compatibility.

Indhold

1 Abstract	i
2 Forord	1
3 Projektstyring	2
3.1 Rollefordeling	2
3.2 Tidsplan	2
3.3 Projektmappe	2
4 Problemidentifikation	3
4.1 Idegenerering	3
4.1.1 Mindmap	3
4.1.2 Lyskurven	3
4.1.3 Identificering af nøgleproblem	3
5 Problemanalyse	5
5.1 Problemtræ	6
5.2 Kvalitativ metode	7
5.3 HV-modellen	7
6 Produktprincip	8
6.1 Målgruppe	8
6.2 Kravspecifikation	8
6.3 Konkurrenter	8
7 Produktudformning	9
7.1 Overordnet	9
7.2 Kodestack	9
7.2.1 Framework - React Native	9
7.2.2 Sub-Framework - Expo	9
7.2.3 Runtime - Node.js	9
7.2.4 Database - MongoDB*	9
7.2.5 Programmeringsprog - Typescript	9
7.3 Brugergrænseflade	10
7.3.1 Konceptdesign	10
7.3.2 Design	10
7.4 Kodegennemgang	10
7.4.1 Filstruktur	10
Litteraturliste	12
Appendiks	12
A Projektbeskrivelse	13
B Konceptart	A15

2 Forord

Vi vil gerne rette en tak til Morten Bach Jensen, agronom. Morten hjalp os med vores afgrødeberegner, han hjalp os også med information om sæddosering samt arealallokering, anvendt til beregning.

Vi vil endvidere rette en tak til Jacob Søgaard, Ph.d Video Quality Assessment, som har hjulpet os med viden om hvordan vi møder vores krav om at kunne gemme videoerne på en optimal måde, så videoerne fylder mindst muligt og kvaliteten er højest muligt.

Sidst, men ikke mindst vil vi takke vores lærer, Henrik Neumann Poulsen, for hans støtte og hans gode råd igennem hele projektet.

3 Projektstyring

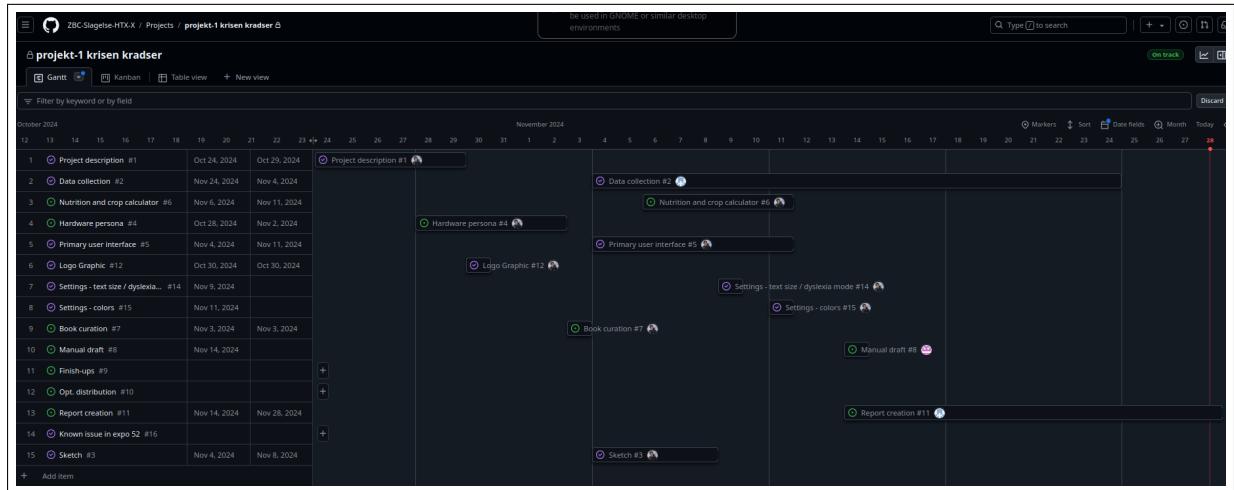
3.1 Rollefordeling

	Roller:
Alexander:	Visionsansvarlig, dokumentarist, strukturering
Andreas:	Konceptartist, informationssøgning
Jeppe:	Produktansvarlig, programør, artistsupervisor, vision

Tabel 1: Viser rollefordelingen for vores projekt.

3.2 Tidsplan

Vores tidsplan er håndteret via GitHubs indbygget funktion, se forneden, dog er den opdateret version her:



Link til aktuel tidsstyring:

3.3 Projektmappe

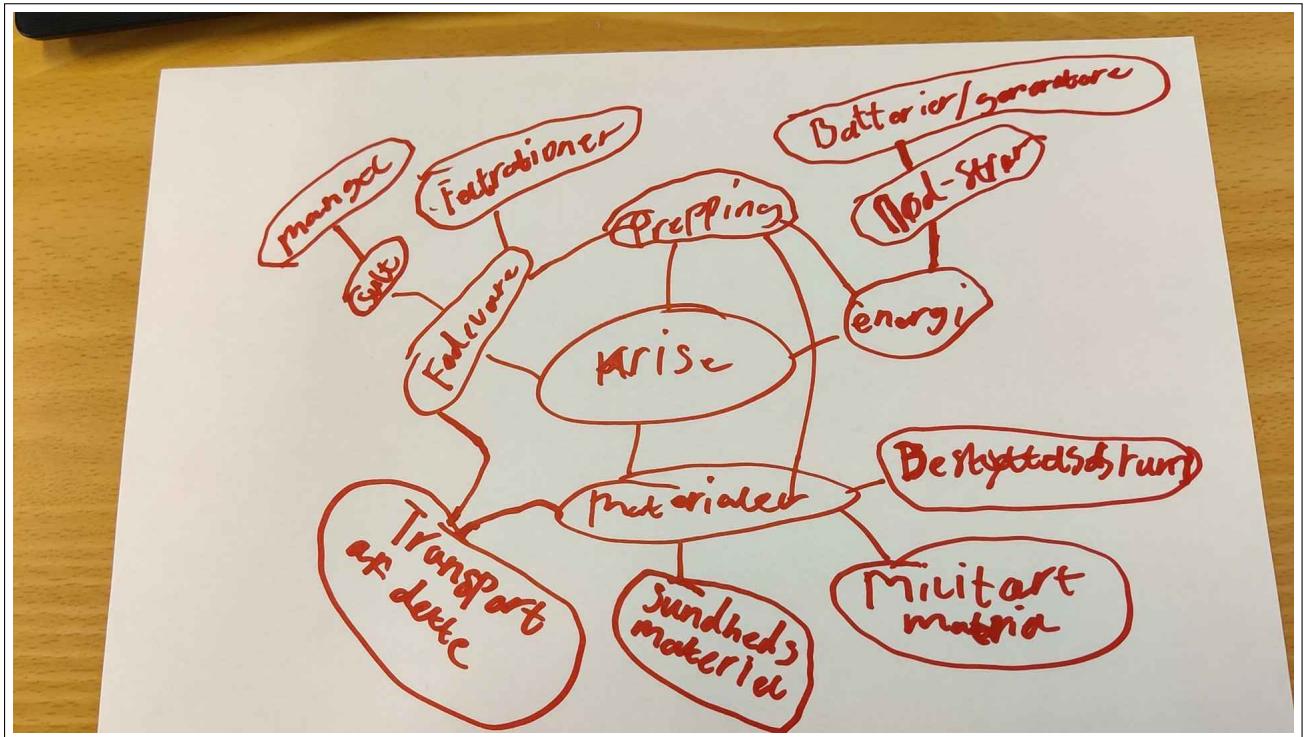
Link: [Vores GitHub fil](#)

4 Problemidentifikation

4.1 Idegenerering

4.1.1 Mindmap

Vi har valgt at lave et mindmap, da dette er en effektiv måde at generere ideer på, og få et overblik over hvilke emner der er relevante at beskæftige sig med.



Figur 1: Viser vores mindmap

4.1.2 Lyskurven

Lyskurven er en metode, man kan bruge til at sortere ideerne i forhold til hvilke der er mest relevante. — Bla bla waffel mere on lyskurvediagrammet —

Fødevarer	Grøn
Beskyttelsesrum	Gul
Nødstrøm	Rød

Tabel 2: Viser et meget abstrakt lyskurvediagram i form af en tabel; det vi anvendte

4.1.3 Identificering af nøgleproblem

Ved identifikation af et nøgleproblem, kan man sortere i sine ideer ved at opstille følgende spørgsmål:

1. Hvorfor er det her interessant?
2. Hvem er det interessant for?

3. Er det noget, vi laver for vores egen fornøjelses skyld?
4. Er det noget, som en bestemt gruppe i samfundet kan have gavn af, eller er det noget, der er til gavn for alle?

** Spørsmålene som vi her gør brug af er hentet fra systimebogen¹

Besvarelsen på disse spørgsmål ser således ud:

1. Krisehåndtering er et interessant emne, da det har en stor betydning for alle i samfundet.
2. Det er relavant for alle.
3. Nej, ideen med produktet er at kunne hjælpe almindelige mennesker med at håndtere kriser, mindre som store.
4. Produktet skulle kunne gavne alle, som kunne stå i en krisesituation.

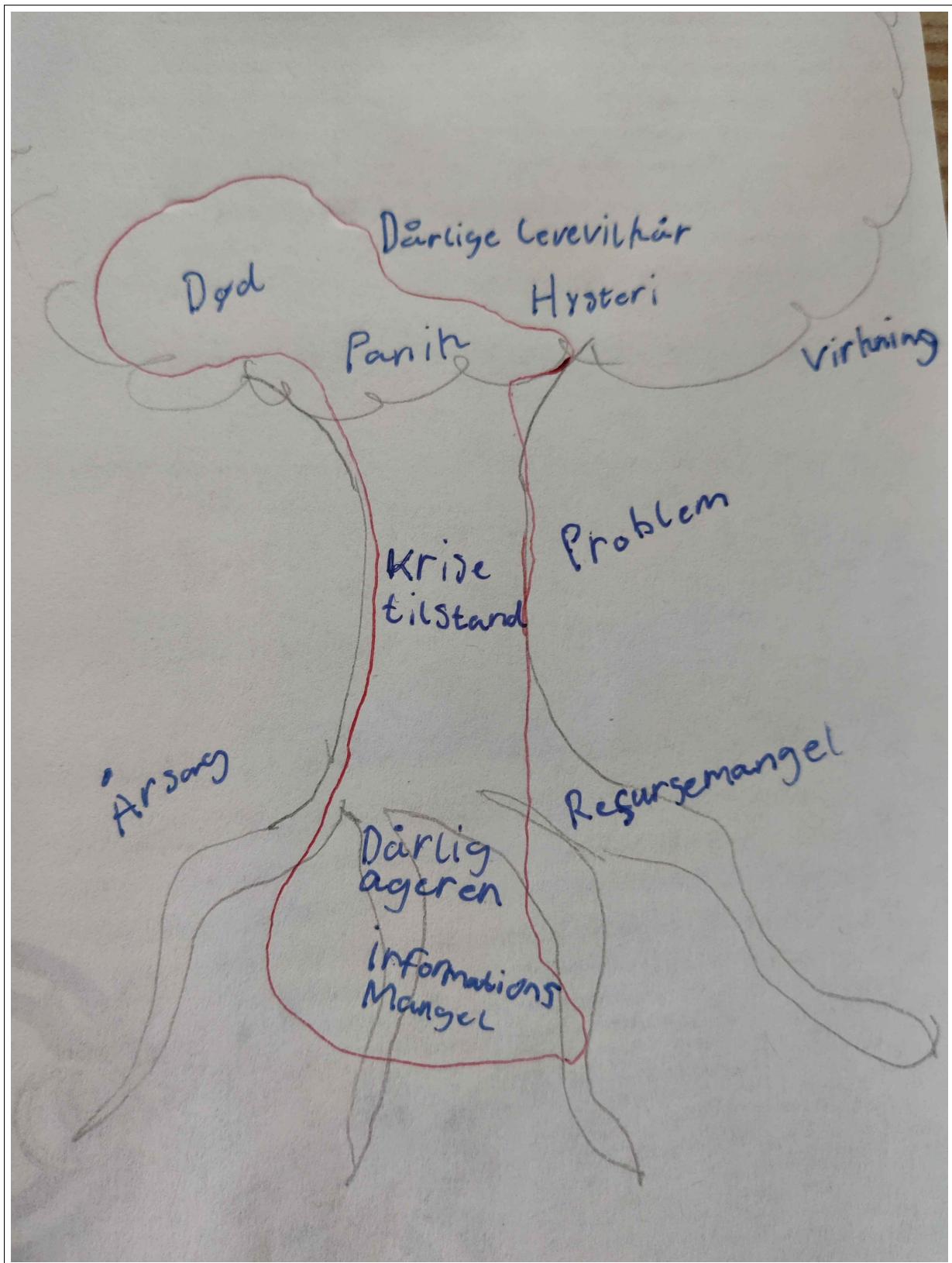
Vi kan nu se, at det mest interessante emne, jf. overstående, er krisehåndtering, da det har en så stor relavans for grupper i samfundet, og det er noget som vi ser der er behov for.

¹[Projektarbejdet](#)

5 Problemanalyse

Problemanalysen tager udgangspunkt i nøgleproblemet krisehåndtering jf projektbeskrivelsen, der kan findes som appendiks ([A](#)) samt kan man læse kapitlet om problemidentifikation ([4](#)).

5.1 Problemtræ



Figur 2: Viser vores problemtræ

5.2 Kvalitativ metode

Beredskabstyrelsen har udarbejdet og sendt en opfordring til alle danskere om kriseparathed, hvor de tydeligøre vigtigheden i at være beredt på at kunne håndtere en eventuel krisesituation. De beskriver i denne, hvordan danskerne skal kunne klare sig selv i 3 døgn, og hvad der skal til for at dette er muligt. I vores husstande, i det private, er disse opfordringer blevet taget meget seriøst. Men hvad gør man, hvis man løber tør for drikkevand, mad eller lignende? Man kan nødvendvis ikke tilgå hjælp via internettet. Vi så her en god mulighed, med vores appudviklingsenskaber, at kunne løse dette problem med at lave en app, der kan hjælpe med at håndtere kriser, både forberedene, på kort sigt og på lang sigt.

5.3 HV-modellen

Ligedan er HV-modellen blevet brugt for at konkretisere, hvilke trin som tages for at udføre vores projekt.

Hvad? - Det skal gøres overskueligt og konkret, hvordan man skal handle i en krisesituation.

Hvorfor? - Det skal gøres, så man kan være beredt i en eventuel krisesituation. Fra et firmasynspunkt er der et stort udbyttepotentiale, da det må forventes, at folk værner om sit og sine næstes liv. Desuden er det sandsynligt, at eftersom beredskabstyrelsen har varslet information om kriseparathed [1], at man via en aftale med staten eventuelt kunne få et subsidium til udarbejdelsen af en sådan applikation, som heri benævnes.

Hvem? - Projektet skal udarbejdes af en nyopstartet virksomhed, her fiktivt, "KEP". Se mere om "KEP-størrelsen under kapitlet om distribution og praksisudførelse ??.

Hvordan? - Det skal gøres ved at lave et program, der gør overstående jf. "hvad?", via en moderne applikationsbyggeløsning, således at denne finde den gyldne vej imellem optimisering og kompatabilitet.

6 Produktprincip

6.1 Målgruppe

Det tilsigtes, at produktet skal være tilgængeligt for alle danskere, da kriser ikke diskriminerer. Det betyder i praksis, at personer, der har diverse handicap, skal være i stand til at kunne produktet, herunder folk, som er ord- eller farveblinde.

6.2 Kravspecifikation

Da det må forventes, at produktet skal kunne anvendes i tilfælde af et nedbrud af internettet, herfor skal produktet kunne fungere uden internetforbindelse dvs. applikationen skal være offline—og da det er planen, at appen skal indeholde videoer, skal disse ligeledes nedhentes. Desuden skal produktet være simpelt og intuitivt at forstå, herunder have ordforklaringer og selve brugergrænsefladen skal være let at navigere rund i og følge nuværende UI-standarder. Produktet skal ligedan være et kompromis mellem design og letvægtighed, herved forstås der, at appen ikke er resursekraævende, således at selv ældre hardware kan tilgå appen.

Herved på listeform:

- Offlinefunktionalitet
- Letvægtighed
- Simpel brugergrænseflade samt ordforklaringer

6.3 Konkurrenter

Der er ikke umiddelbart nogen decideret konkurrent til produktet, især ikke på det danske marked, som henvendes til, men der et utal af videoer og guides om overlevelse i det vilde og diverse beregnere på internettet, men her adskiller det forslættet produkt sig ved at være tilgængeligt uagtet internetforbindelse, mere omfattende og tilgængeligt på dansk.

7 Produktudformning

7.1 Overordnet

Koden er opbygget, således at den kan skrives som en pseudo-webapplikation, sidenhen anvendes en "bro", der gør til en native applikation og kompatibel med de mest almindelige styresystemer, herunder IOS (Apples mobile styresystem) og Android².

7.2 Kodestack

7.2.1 Framework - React Native

React Native er et framework³, der muliggør udvikling af native⁴ applikationer til IOS og Android.

7.2.2 Sub-Framework - Expo

Expo er et framework, der er bygget omkring React Native, der muliggør at køre en pågældende React Native-applikation igennem deres egen platform, Expo Go, hvorfor man ikke behøver at kompile koden, dvs. oversætte programmeringskoden til eksekverbar maskinekode, efter hver ændring, såfremt man vil teste den på en fysisk enhed.

7.2.3 Runtime - Node.js

Node.js er den runtime, der muliggør at applikationen kan køre på en enhed fremfor i en browser, da programmeringssproget JavaScript oprindeligt var designet til at køre udelukkende i et browser-miljø.

7.2.4 Database - MongoDB*

Ideen var oprindeligt at have en meget let applikation, der kunne nedhentes fra et styresystems nativ app-bibliotek, hvorefter denne ville prompte brugeren til at nedhente ekstrapakker, fx videoer og manualer, fra vores egen server, som skulle være administreret via MongoDB. MongoDB er et program, som tillader en at lave og administrere enNoSQL-database, hvilket er en mere fleksibel database end traditionelle SQL-databaser.

7.2.5 Programmeringsprog - Typescript

Typescript er et programmeringssprog udviklet af Microsoft, som bygger på JavaScript. Typescript bruger stærke datatyper, hvilket vil sige, at datatypen angives per data. Typescript anvendes i denne applikation, fordi det er et kompilersprog, hvilket vil sige, at fejl kan blive fanget, når koden kompileres, fremfor i runtime, mens programmet kører, hvilket er en stor fordel i programmeringsprocessen, da det hindrer, at der kommer oversete fejl i koden.

²Selvsamme teknik anvendes af højprofilerede virksomheder, såsom Facebook, Discord og Tesla. Læs mere her: reactnative.dev

³Et framework er en samling af biblioteker, der gør det lettere at skrive kode på en standardiseret måde, hvori en masse valg er taget på forhånd

⁴Native betyder at applikationen kører direkte på enheden

7.3 Brugergrænseflade

Brugergrænsefladen er det, brugeren oplever, når han interagerer med applikationen. Brugergrænsefladen er blevet tegnet i Figma, et program, som er speciallavet til at konstruere brugergrænseflader, bl.a. har den prælavede elementer, fx knapper og tekstinputfelter, desuden kan dette gøres interaktivt.

7.3.1 Konceptdesign

Jf. appendiks INSERT FIGURE er følgende skitser blevet udarbejdet. Herefter er disse blevet implementeret i selve applikationen. Brugergrænseflade består primært af en bjælke, der er placeret i bunden af skærmen, hvorfra brugeren kan navigere mellem forskellige sider.

7.3.2 Design

7.4 Kodegennemgang

Koden som bygger applikationen, vil blive gennemgået nedenfor.

7.4.1 Filstruktur

Måden hvorpå filerne er struktureret, er altafgørende for kodens funktionalitet og læsbarhed. Da frameworkt React Native, kigger efter spicifikke filstrukturer, til at køre og vise den dertilhørende kode det korrekte sted.

```
/  
├── app  
├── assets  
├── components  
├── constants  
├── data .2 hooks  
├── node_modules  
├── scripts  
├── .gitignore  
└── app.json  
    ├── babel.config.js  
    ├── eas.json  
    ├── package-lock.json  
    ├── package.json  
    └── tsconfig.json
```

Figur 3: Top-level filstrukturen for applikationen. Mapperne er farvet blåt

Måden hvorpå brugen navigerer applikationen, er ved at bruge vores navigationsbar, som er placeret i bunden af skærmen. Filstrukturen som viser navigationsbaren, ser således ud i kodebasen:

```
/  
├── app  
├── assets  
├── components  
├── constants  
├── data .2 hooks  
├── node_modules  
├── scripts  
└── .gitignore  
├── app.json  
├── babel.config.js  
├── eas.json  
├── package-lock.json  
├── package.json  
└── tsconfig.json
```

Figur 4: Top-level filstrukturen for applikationen. Mapperne er farvet blåt

Litteraturliste

1. Beredskabstyrelsen. *Forberedt på krise* https://www.brs.dk/globalassets/brs---beredskabsstyrelsen/dokumenter/forberedt--forberedt_paa_kriser_lowres-.pdf (2024).

Appendiks

A Projektbeskrivelse

Projektbeskrivelse af projekt 1 ”Krisen kradser” i faget Teknologi B

Alexander Knudsen, Andreas Jensen og Jeppe Bech

29. oktober 2024

1 Formalia

1.1 Gruppесammensætning

Projektet, som heri beskrives, er udarbejdet af følgende personer: Alexander Knudsen, Andreas Jensen og Jeppe Bech (2. X).

1.2 Tema

Overordnet: Krisen kradser

▀ Underemne: Fødevarer

2 Produktbeskrivelse

Produktet, kulminationen på projektet, er en offline softwareprogram med diverse delprogrammer til krisehåndtering, herunder en ernæringsberegner, der også indeholder modstykke, der foretager anbefalinger om afgrødevalg og allokering af areal til disse.

Endvidere indeholder programmet også manualer med information til afljælpning af diverse nødsituationer samt praktisk information vedrørende daglig ageren, fx istandsættelse og anvendelse af radio, reparation af bil, isolering af hus, et cetera.

3 Tidsplan

Tidsplanen og tidsstyringen foregår via GitHubs indbyggede funktionlaitet, dog tilstræbes det som udgangspunkt, at projektet foregår over fem faser, dog er dette fleksibelt og kan korrigeres undervejs:

1. Dataindsamling og skitseudarbejdelse (Projektesgrundlag) - varighed (1 uge)
2. Brugeroverflade samt udarbejdelse af hardwarepersona (Produktudvikling) - varighed (3 dage)
3. Ernærings- og afgrødeberegner (Produktudvikling) - varighed (1 uge og 4 dage)

4. Manualuddrag samt bogtilvalg til henvisning, finpudsning og evt. *visualiser* - varighed (1 uge)
5. Evt. distribution og afrappotering - varighed (1 uge)

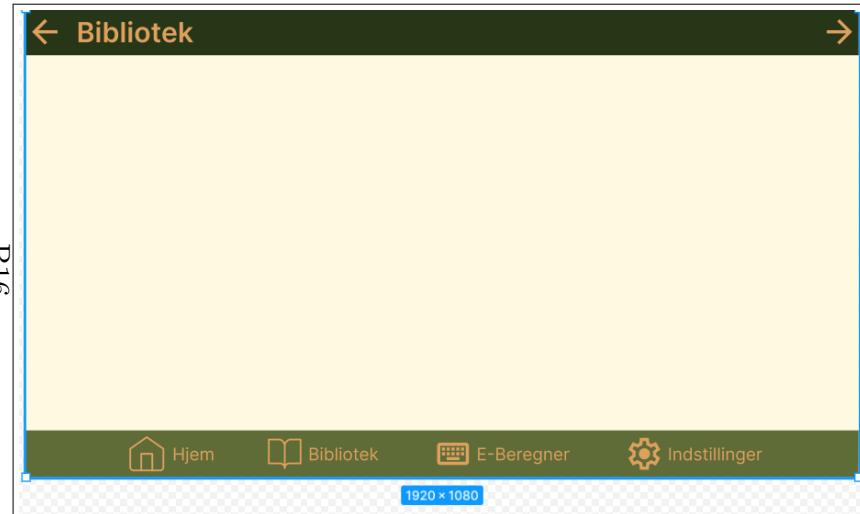
B Konzeptart

Samling af konceptskitser

Alexander Knudsen, Andreas Jensen og Jeppe Bech

5. december 2024

1 Navigationsbjælkekoncept



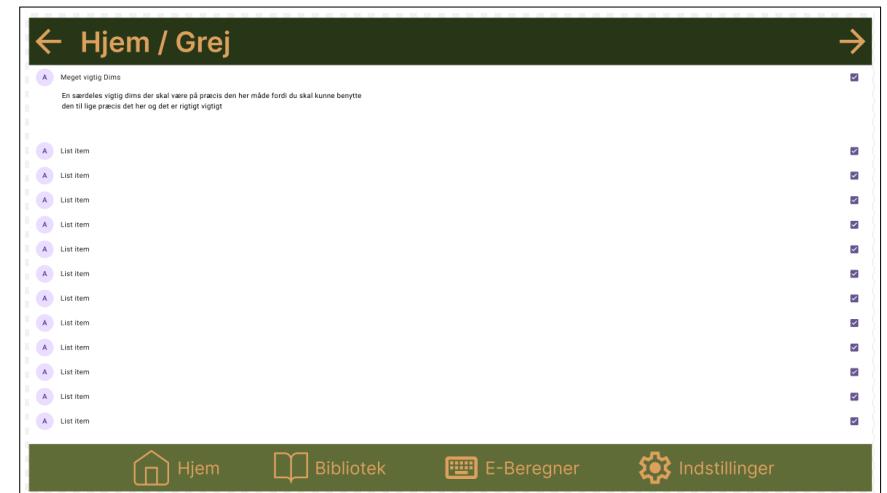
Figur 1: Viser Navigationsbjælkekonceptet

2 Manualbibliotekskoncept



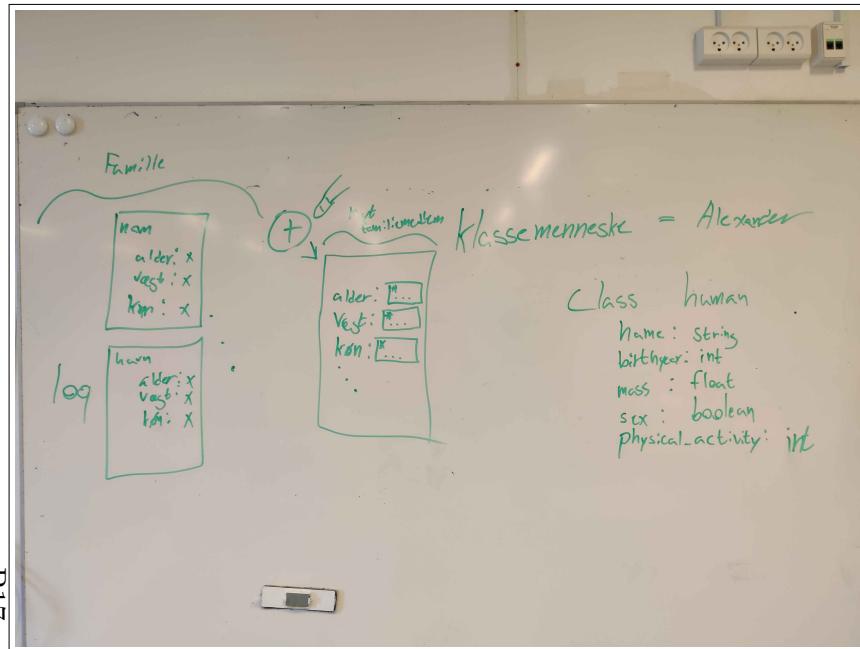
Figur 2: Viser manualbibliotekkonceptet

3 Grejlistekoncept



Figur 3: Viser grejlistekonceptet

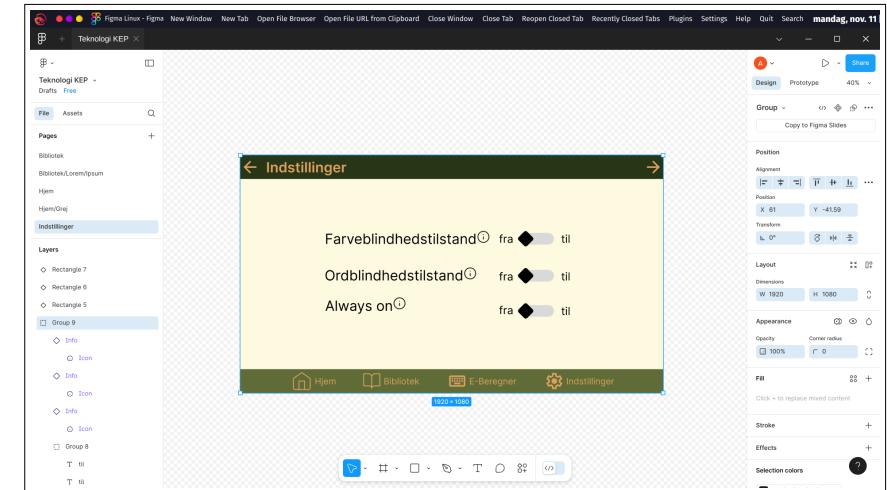
4 Ernæringsberegnerkoncept



B17

Figur 4: Viser ernæringsberegnerkonceptet

5 Tilgængelighedsmenukoncept



Figur 5: Viser tilgængelighedsmenukonceptet