

---

# Projekt - studiemiljø

Forårssemester 2024

---

Jeppe Bøgeskov Bech  
[jepp9920@zbc.dk](mailto:jepp9920@zbc.dk)

Alexander Schade Knudsen  
[alex245h@zbc.dk](mailto:alex245h@zbc.dk)

Andreas Jensen  
[andr328q@zbc.dk](mailto:andr328q@zbc.dk)

David Rasmussen  
[davi5621@zbc.dk](mailto:davi5621@zbc.dk)



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

1. x

ZBC Handels- og Teknisk gymnasium Slagelse

Akademisk år 2023-2024

15. maj 2024

# 1 Abstract

This report encompasses a full description of the Lectio application suite which is a complete IT-solution directed at the Danish High Schools. It allows for centralised data sharing in regards to homework, assignments and lecture management. It provides all of this in an asthetically pleasing shell which utilises state of the art design philosophy and technology behind the scenes to provide the user with a pleasurable experience which both respects the user and his time.

The system is two-folded in that it both encompasses software and hardware—hardware in terms of a door control system which utilises RFID-technology and advanced absence registration.

The document below is written in Danish and both documents and explains design choices, functionality and the software suite.

## 2 Om valg af formidling, informationsteknik og orientering

Denne rapport henvender sig til de relevante faglærer og dokumenterer teknologiprojektet i forårssemesteret.

Forneden gennemgås strukturen samt nogle af designvalgene bag opsætningen af rapporten.

Dokumentet er skrevet med fonten *Latin Modern*, grundet dens kompatibilitet med diverse matematik, sprog og symboler, med matematik i skriftstørrelsen 12pt. Præliminærssiderne er pagineret med romertal, brødteksten med arabiske tal og appendikssiderne med dets bogstav samt de arabiske tal.

Dokumentet er typesat via L<sup>A</sup>T<sub>E</sub>X, et markup-sprog, da det tillader for utroligt smukke dokumenter, nem numerering samt administration af figurer, tabeller, bibliografier og appendikser. T<sub>E</sub>X-kodefilerne kan tilgås via GitHub, ligedan med kodedelen af projektet: <https://github.com/ZBC-Slagelse-HTX-X/Teknologi-project>.

# Indhold

<b>1 Abstract</b>	i
<b>2 Om valg af formidling, informationsteknik og orientering</b>	ii
<b>3 Opgavevalg</b>	1
3.1 Formål og opgavekrav . . . . .	1
3.2 Oplæg . . . . .	1
<b>4 Projektstyring</b>	2
4.1 Rollefordeling . . . . .	2
4.2 Tidsplan . . . . .	3
<b>5 Inledning</b>	4
<b>6 Problemanalyse</b>	5
6.1 Problemtræ . . . . .	5
6.2 Kvalitativ metode . . . . .	7
6.3 HV-modellen . . . . .	7
6.4 Envidere kritik af Lectios designfilosofi . . . . .	7
6.5 Problemformulering . . . . .	8
<b>7 Produktprincip</b>	9
7.1 Målgruppe . . . . .	9
7.2 Kravspecifikation . . . . .	9
7.3 Kort om konkurrenter . . . . .	9
7.4 Løsningsforslag . . . . .	9
7.4.1 Lectio rework . . . . .	9
7.4.2 RFID-Løsning . . . . .	11
<b>8 Produktudformning</b>	13
8.1 Lectio rework . . . . .	13
8.1.1 Overordnet . . . . .	13
8.1.2 Kodegennemgang . . . . .	13
8.2 Bookingsystem . . . . .	21
<b>9 Produktionsforberedelse</b>	22
9.1 Website-hosting . . . . .	22
9.2 RFID-løsning / låsemekanismesystem . . . . .	22
9.2.1 Fremtidig Plan . . . . .	22
9.2.2 Produktion . . . . .	23
9.3 Teknologianalyse . . . . .	23
<b>10 Evaluering, vurdering og konklusion</b>	24
10.1 Test af produkt . . . . .	24
10.2 Teknologivurdering . . . . .	24
10.2.1 Konsekvensvurdering . . . . .	24
10.3 Procesevaluering . . . . .	24
10.4 Konklusion . . . . .	24

<b>Bibliografi</b>	<b>25</b>
<b>Appendiks</b>	<b>25</b>
<b>A Projektbeskrivelse</b>	<b>26</b>
<b>B Logbog</b>	<b>30</b>
<b>C Arbejdsskitser</b>	<b>33</b>
<b>D Modulbrikkoden</b>	<b>37</b>
<b>E Skemakomponentkode</b>	<b>D40</b>
<b>F Datahåndteringssystem</b>	<b>D45</b>
<b>G Eksempel på beskedfunktionalitet</b>	<b>F49</b>

## **3 Opgavevalg**

### **3.1 Formål og opgavekrav**

Teknologiprojektet beskrevet heri omhandler HTX' studiemiljø, hvortil der er tre oplæg.

### **3.2 Oplæg**

I dette teknologiprojekt tages der udgangspunkt i oplæg nr. 2, der omhandler "Studiemiljø og de fysiske rammer". Vi valgte dette oplæg, da vi besidder kompetencer, der gør, at vi kan få et godt produkt ud af dette ift. softwareudvikling, men også i forhold til manipuleringen af resurser til at lave et fysisk produkt.

Dette oplæg virker også til at være det mest oplagte til at udforme et reelt produkt af fysisk og seriøs karakter.

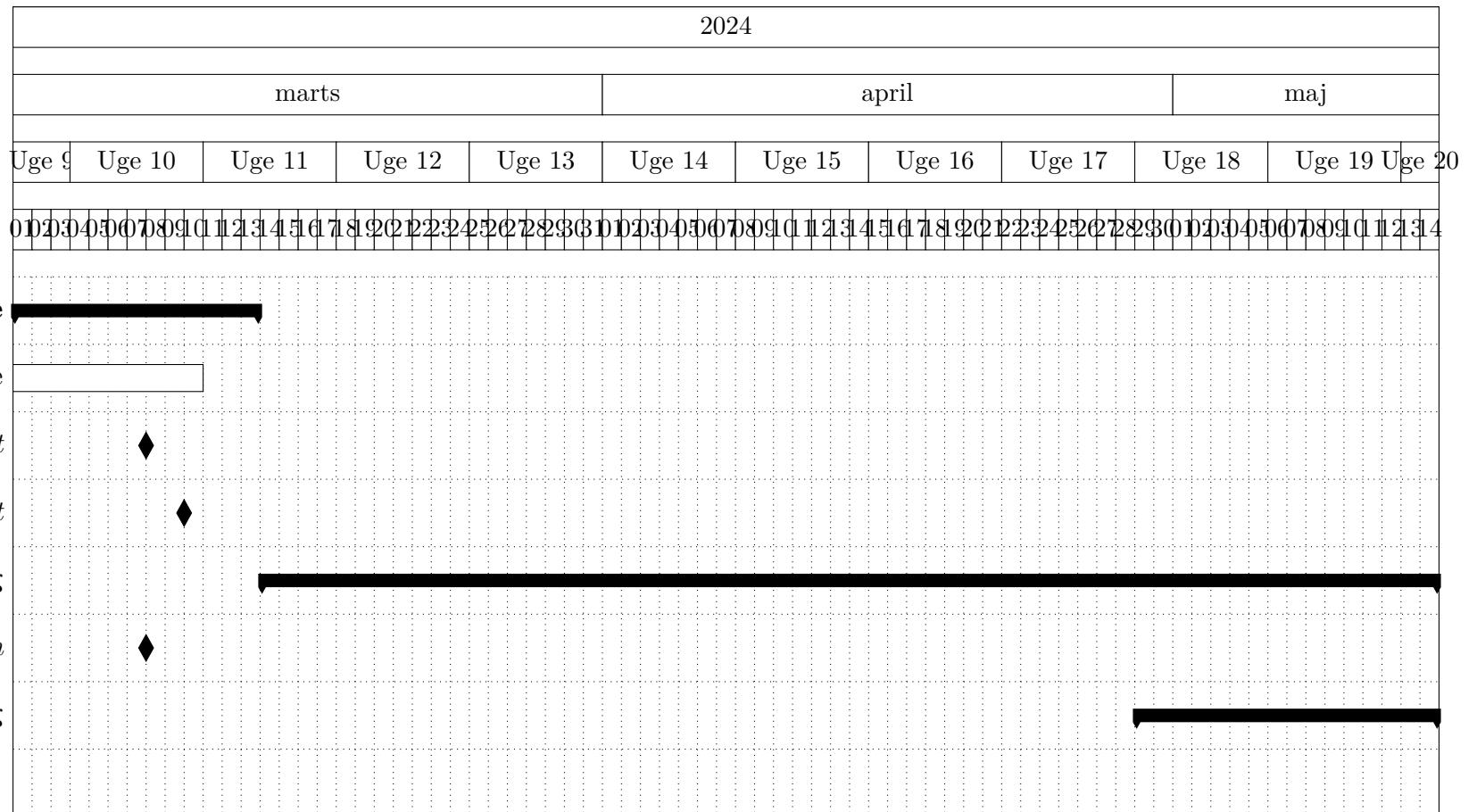
## 4 Projektstyring

### 4.1 Rollefordeling

Navn	Ansvarsområde
Jeppe	Kodning, skitsering, GitHub, vision
Alexander	L <sup>A</sup> T <sub>E</sub> X & opstilling & strukturering, vision
Andreas	Hardware
David	Logging

Tabel 1: Viser gruppemedlemmernes henholdsvis ansvarsområder.

## 4.2 Tidsplan



Figur 1: Viser Gantt-Diagram over vores foreløbige tidsplan. Denne er korrigeret undervejs, se appendiks.

## 5 Inledning

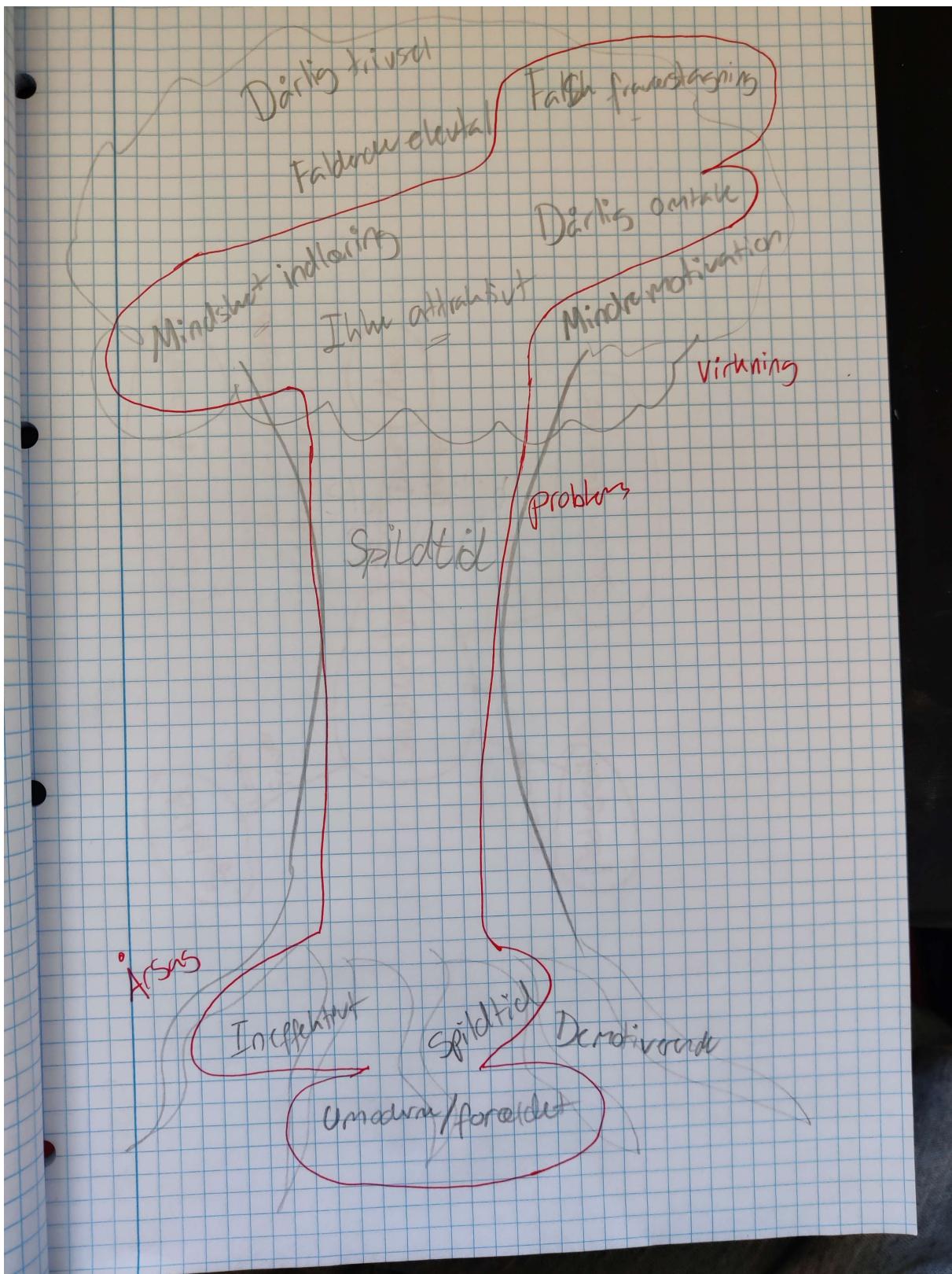
Jf. problemanalysen, problemformuleringen [6.5](#), er følgende vores problemstilling: » Det er et problem, at Lectios website er forældet, da dette resulterer i, at den ikke effektivt kan overlevere information til sine forbrugere, ej heller overskueligt intergeres med jf. problemanalysen [\(6\)](#), hvorfor den ikke er berettiget i sin eksistens jf. Lectios eget informationsark, hvor de eksplisit udpenslerer dette som Lectios bærende hovedformål[\[1\]](#).«, hvorfor projektet tager udgangspunkt i dette.

## **6 Problemanalyse**

Problemanalysen tager udgangspunkt i nøgleproblemet spildtid jf. projektbeskrivelsen i appendikssektionen ([A](#)).

### **6.1 Problemtræ**

Forneden har vi fremstillet et problemtræ, der viser nogle af årsagerne til tidsspild og nogle af virkningerne.



Figur 2: Viser problemtræet, der blev anvendt i problemanalysen.

Et af de ineffektive systemer, vi kom i tanke om, er Lectio-applikationen, hvorefter vi har informationssøgt om emnet. Det var ikke muligt at finde nogle reelle artikler om tidssplid med Lectio-applikationen. Da har vi forhørt os kvalitativt og snakket med flere

medstuderende og flere underviserer (interessenter) som benytter sig at Lectio i deres dagligdag.

## 6.2 Kvalitativ metode

Forneden er vores overordnet indtryk af, hvordan de folk, vi har snakket med, opfatter lectio: De har alle udtalt sig om Lectios unødvændigt komplicerede designvalg, forældede udseende og mangler. Vi snakkede bl.a. Med en underviser som udtrykte sin frustration over at det nuværende Lectio kun kan uploadet et dokument ad gangen. Dette fandt vi meget mærkeligt, da vi har en udvidet viden om webudvikling, og ergo ved vi at sådan en funktion (multiupload) er utrolig nem at implementere og stod tilbage uforstående for, at Lectio ikke havde mulighed for så basal funktionalitet som det. Det kan også hurtigt tage fx. 10 sekunder at navigere hen til en fil og derefter uploadet den, så hvis man vil uploadet fx. »20 filer«, så tager dette altså 200 sekunder tilsvarende til 3 minut og 20 sekunder.

## 6.3 HV-modellen

Herefter er HV-modellen i sammenspil med den kvalitative undersøgelse i forrige kapitel, blevet anvendt til at få en ide om handlingsplanen, forbrugernes mangler og dermed potentielle produktbehov:

**Hvad?** Det man skal gøre er altså at implementere basale QOL-features (Quality-of-life; rare features at have) ved at lave et rework af Lectio. På nuværende tidspunkt er vi ikke klar over, hvem der maintainer Lectio-applikationen, ej heller hvilke tiltag de gør for at modernisere denne.

**Hvorfor?** Det skal gøres, så man i fremtiden kan spare tid og tankekraft ved at gøre Lectio til en bedre IT-løsning; et system, der gør det attraktivt at anvende til at centralisere informationsdeling, således at brugere ikke søger afsides og udliciterer opgaven til 700 andre tjenester samt gør det nemtoverskueligt for brugeren at orientere sig; hvilket gør at læringen kan komme i fokus, fremfor IT-stads.

**Hvem?** Hvem bør gøre det? Det er selvfølgelig Lectio

**Hvordan?** De bør gøre det ved at anvende moderne webapplikationsbyggeteknik; ændre deres techstack og designfilosofi.

## 6.4 Envidere kritik af Lectios designfilosofi

Alle knapperne er små, og der er for mange af dem, det er for komplekst og der er meget redundans; mange måder, hvorpå man kan opnå samme resultater. Farvevalget er også grimt; det ser visuelt forældet ud, hvilket ikke just inspirerer en gejst host kunden. Endvidere kan Lectio overhovedet ikke scalerer, altså når man ændre skærmstørrelsen forbliver Lectio samme størrelse. Dette er et rød flag for hjemmesider da korrekt websidedesignfilosofi siger at hjemmesider skal være responsive.

## 6.5 Problemformulering

Det er et problem, at Lectios website er forældet, da dette resulterer i, at den ikke effektivt kan overlevere information til sine forbrugere, ej heller overskueligt intergeres med jf. problemanalysen (6), hvorfor den ikke er berettiget i sin eksistens jf. Lectios eget informationsark, hvor de eksplisit udpenslerer dette som Lectios bærende hovedformål[1].

## **7 Produktprincip**

### **7.1 Målgruppe**

Efter som det nuværende Lectio bliver anvendt af en stor del af landets gymnasier, er vores målgruppe derved alle landets gymnasier – nemlig også potentielle kunder (gymnasier) der muligvis ville være interesseret i at anvende vores Lectio applikation.

### **7.2 Kravspecifikation**

Kravene for vores applikation er følgende:

- Applikationen skal have alle de samme core-features som det originale Lectio
- Applikationen skal være let overskuelig og nemt orienterbart for brugerne
  - Applikationen skal således være simplere end sin forløber
- Applikationen skal indeholde et Lokale-Booking-System/Fraværsregistreringssystem/Dørssystem
  - Applikationen skal have forbedret fraværsregistrering
  - Applikationen skal give alle eleverne et digitalt persona i sammenspil med fraværsregistreringssystemet.
  - Et komplet dørssystem som låses/låses op med RFID kort (på sigt ens mobiltelefon).

### **7.3 Kort om konkurrenter**

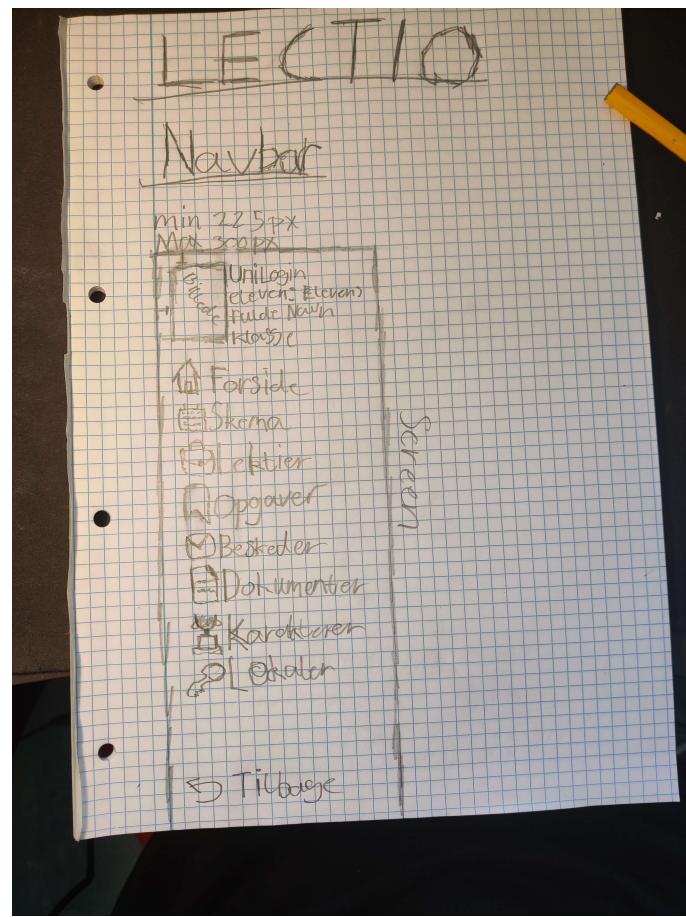
Lectio har ingen umiddelbare konkurrenter, vi kender, udover Aula, men det henvender sig til folkeskoler og har ikke envidere fildelingsinfrastruktur. Vi tror også, at problemet med Lectio stammer fra det faktum, at Lectio har et de-facto monopol på IT-løsninger målrettet gymnasier.

### **7.4 Løsningsforslag**

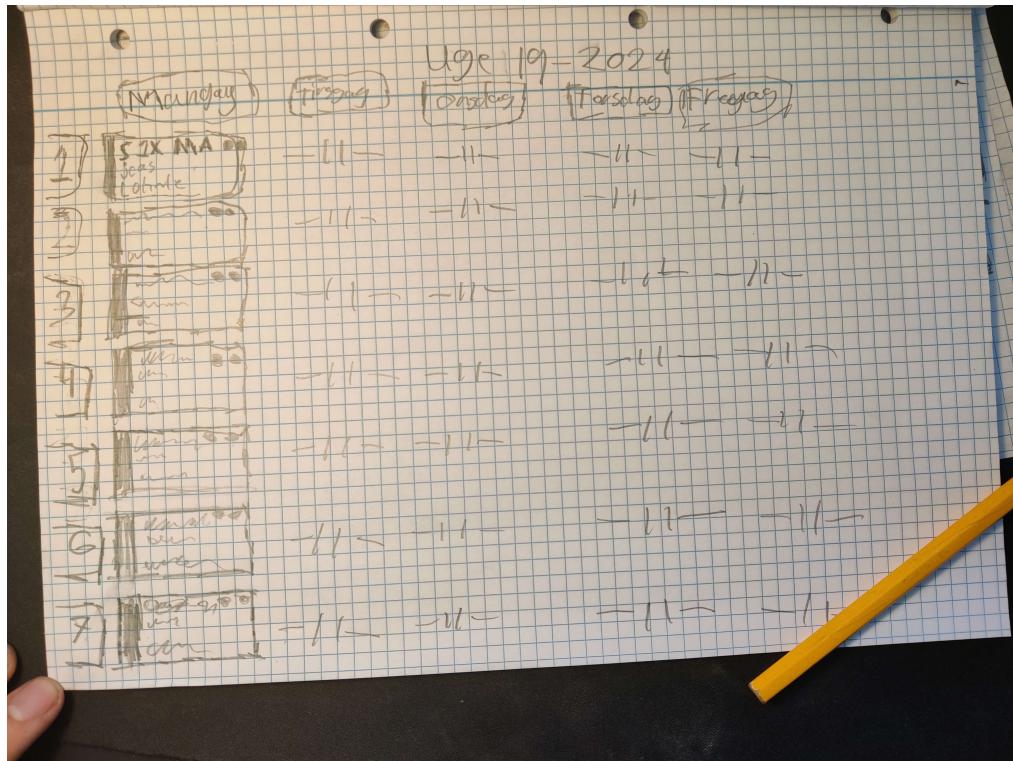
#### **7.4.1 Lectio rework**

For at løse problemet med det forældede Lectio har vi besluttet at lave et komplet rework (dvs. genopbyggelse af softwaren, men med samme kernefunktionalitet) af Lectio applikationen.

Vi har udarbejdet skitser for at kvalificerer vores produktudformning. Skitserne viser de vigtigste sider af vores Lectio rework.



Figur 3: Viser Lectioreworkets navigationsbarkomponent - bjælken, brugeren intergererer med

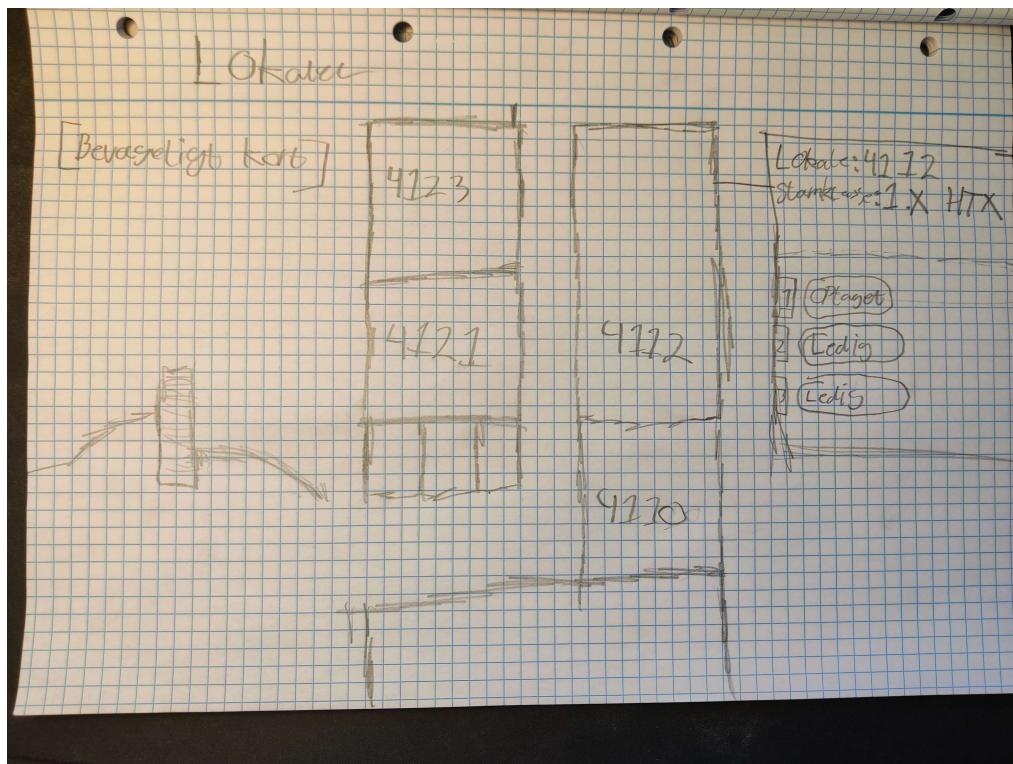


Figur 4: Viser Lectioeworkets skema side

De resterende skitser, vi har udarbejdet, fremgår af appendikssektionen C.

#### 7.4.2 RFID-Løsning

For at tage hånd om problemet med spildtid i form af fraværstagning, fejlagtig fraværstagning m.v. har vi valgt at lave et dør/fraværssystem, som når man om morgnen møder ind, scanner man sig ind med sit ID-kort eller sit digitale ID-kort med sin mobiltelefon på fraværssregisteringssystemet som er i klassen. Lokale-Booking-Systemet fungerer (i ideen—se evalueringen 10) i sammenspil med den nye Lectio applikation, så kan igennem applikationen booke et lokale også med sit ID-kort låse lokalet op i det tidsrum man har booket det pågældene lokale. Web løsningen ses skitseret forneden:



Figur 5: Viser Skitse af Booking siden på samme webapplikation som Lectio Reworket

# 8 Produktudformning

## 8.1 Lectio rework

### 8.1.1 Overordnet

Lectio-applikationen er skrevet i et populært framework, Next.js, som er en overbygning på React-biblioteket til node.js, der tillader at man kan køre javascript på en server i stedet for i Chrome V8-motoren, som kun tillader at man køre javascript clientsided fremfor serversided.

React er et library for user interfaces, der tillader UI-komponenter, der i sammenspil med HTML, udgør en brugeroverflade. HTML står for »HyperText Markup Language«, der er standard markupsproget til alle hjemmesider–alt indhold, der serveres på internettet. Et markupsprog, fx. L<sup>A</sup>T<sub>E</sub>X, som dette dokument er skrevet i, i modsætning til programmeringssprog, anvendes til at overlevere informationer til mennesker–programmeringssprog anvendes til at give instrukser til computere.

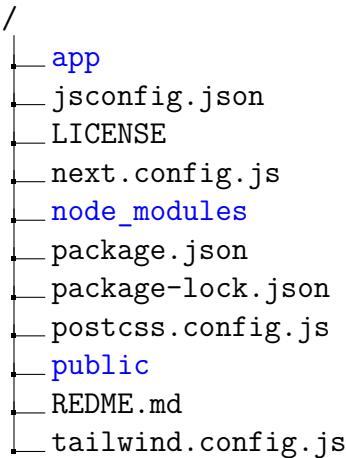
Selve renovering er rent faktisk en renovering i den forstand, at den reelle hjemmeside er blevet forbedret eller fornyet, men derimod er den blevet genopbygget fra bunden af dog med henblik på at bevare den samme funktionalitet. Af den grund er det en mere korrekt betegnelse at kalde det et »rework«.

### 8.1.2 Kodegennemgang

Forneden er en komplet gennemgang af koden til Lectio-reworket.

En filstruktur behøver æstetisk, letoverskuelig og forståelig både i menneske- og computerterminer. Således er der både de prædefinerede valg og de mere artistiske valg.

Et Next.js-projekt (v.14) benytter deres nyligt lancerede app router-filstruktursystem, således skal filer struktureres på følgende vis: man har »Top-level files«, der bruges til applikationkonfiguration, administration af filafhængigheder m.m.[2]



Figur 6: Top-level filstrukturen for lectio-reworket. Mapperne er farvet blåt

Forneden gennemgås top-level filerne og mapperne efter hensyn til forståelsen, der fremgår af figur (6)

**next.config.js** er en konfigurationsfil for Next.js, der tillader, at man konfigurerer sprogets funktionalitet.

**LICENSE-filen** er en fil, der indeholder projektets licens og brugsrettigheder, og da vores er et open source-projekt (open source vil sige, at alle kan tilgå det "gratis"), så bruges MIT-licensen, der tillader al brug af materialet til alle formål af alle individer.

**jsconfig.json** er en projektafhængighed for javascriptsproget, og den indeholder kompileringsvariabler, få.

**tailwind.config.js.** er en konfigurationsfil, der konfigurerer CSS. CSS er et understøttende design systemssprog til HTML, så alt der har med design at gøre, dvs. tekststørrelser, farver mv. "styles" i CSS. I Lectio.v2-applikation anvendes Tailwind-CSS, et framework til CSS, til at style dette; det er lettere overskueligt, hurtigere og standardiseret. Se forneden kode og rendering<sup>7</sup>:

```
1 <div className="bg-[#9CCEFF]
2   border-l-[10px]
3   border-l-[#1E90FF]
4   p-4
5   rounded-lg
6   max-w-[275px]
7   max-h-[88px]
8   flex
9   justify-between
10  cursor-pointer
11  hover:opacity-80
12  group">
13
14 <div className="flex flex-col text-[#0D3F70] justify-center">
15   <span className="font-bold text-xl">S 1x { args.subject }</span>
16   <span>{ args.teacher }</span>
17   <span>{ args.room }</span>
18 </div>
```



Figur 7: Viser en modulbrik m. tilhørende CSS-kode (notér at dette ikke er den komplette kode [tjek bilag D] til brikken, dog godt nok for at illustrere vores pointe)

**app-mappen overordnet** indeholder reworks reelle koder, i modsætning til de andre filer, der primært konfigurer koden og den gør kørbar. Neden for ses hvordan vi har struktureret app-mappen.

```

app
├── page.jsx
└── layout.jsx
  └── dashboard
    ├── layout.jsx
    └── page.jsx
  └── lokaler
    └── page.jsx
  └── skema
    ├── page.jsx
    └── skema.json
    └── [slug]
  └── ui
    ├── dashboard
    ├── navbar
    └── skema
    └── globals.css

```

Figur 8: Filstrukturen for app mappen. Mapperne er farvet blå

I Next.js, er alle mapper i /app-mappen, det er et subdomain, når det indeholder en »page.jsx«-fil. Men hvad er et subdomain? Før vi kan forstå det, skal vi se på, hvad er en URL? URL står for Uniform Resource Locator. En URL er struktureret med først en protokol, som for web ressourcer er enten HTTP (Hypertext Transfer Protocol) eller HTTPS (HTTP Secure). Andre protokoller inkludere f.eks. FTP som er en »File Transfer Protocol« efter protokollen kommer domænenavnet. Domænenavnet referer til en ip-adresse. Det vil sige, at når man søger efter et domænenavn, søger man i en DNS-server (Domain Name System) og finder den tilknyttede ip-adresse, som sender den data, som hjemmesiden består af til browseren. Efterfølgende kan der være subdomains der er alt efter domainenavnet plus et /.



Figur 9: Viser opbygningen af en URL

**package.json** indeholder modulafhængigheder, der er hentet via NPM (Node package manager), der udnytter node.js. Forneden er et udsnit fra dette, der viser »dependencies«:

```
1 "dependencies": {  
2   "@heroicons/react": "^2.0.18",  
3   "@vercel/postgres": "^0.5.1",  
4   "bcrypt": "^5.1.1",  
5   "current-week-number": "^1.0.7",  
6   "dotenv": "^16.3.1",  
7   "heroicons": "^2.0.18",  
8   "next": "14.0.3",  
9   "react": "^18",  
10  "react-dom": "^18",  
11  "react-draggable": "^4.4.6"  
12},
```

Mod venstre kan man se navnet på NPM-pakken. Efter kolonnet til højre ses versionen af denne. Næsten alle dem, der fremgår, bliver udnyttet, dog er der få som blot er resultatet af skabelonbrug. Denne skabelon er tiltænkt til fremtidig brug, så den kan anvendes til flere formål. Skabelonen kan også tilgås via GitHub.

Her er en kort gennemgang af nogle af pakkerne, der er hentet via NPM, som vi bruger, der tilføjer ekstra funktionalitet:

- heroicons - Anvendes forskellige steder i koden, hvor ikoner anvendes. Heroicons fungerer således, at ønskede ikoner kan importeres fra heroicon-biblioteket, hvorefter de kan indsættes i web-applikationen
- current-week-number - Anvendes til at fremkalde det nuværende ugetal via en applikation programm. Det i vores tilfælde til at indsætte ugetallet i skemabrikken.

**app-mappen** Vi startede med at gennemgå overordnet app-mappen ([8.1.2](#)), her er en dybdegående genemgang. App-mappen består af subdomænemapper og en UI-mappe (user interface).

UI-mappen består af komponenter, fx. navigationsbaren, der anvendes på diverse web pages. Heraf er nogle af disse komponenter konsekvente på alle webpagesne, hvorfor det ikke giver mening at genskrive dem i hvert subdomæne. Disse konsekvente komponenter har ophav i root-layout.jsx mappen.

```

app
├── page.jsx
└── layout.jsx
    ├── dashboard
    ├── lokaler
    └── skema

```

Figur 10: Filstrukturen for app mappen. Mapperne er farvet blåt

```

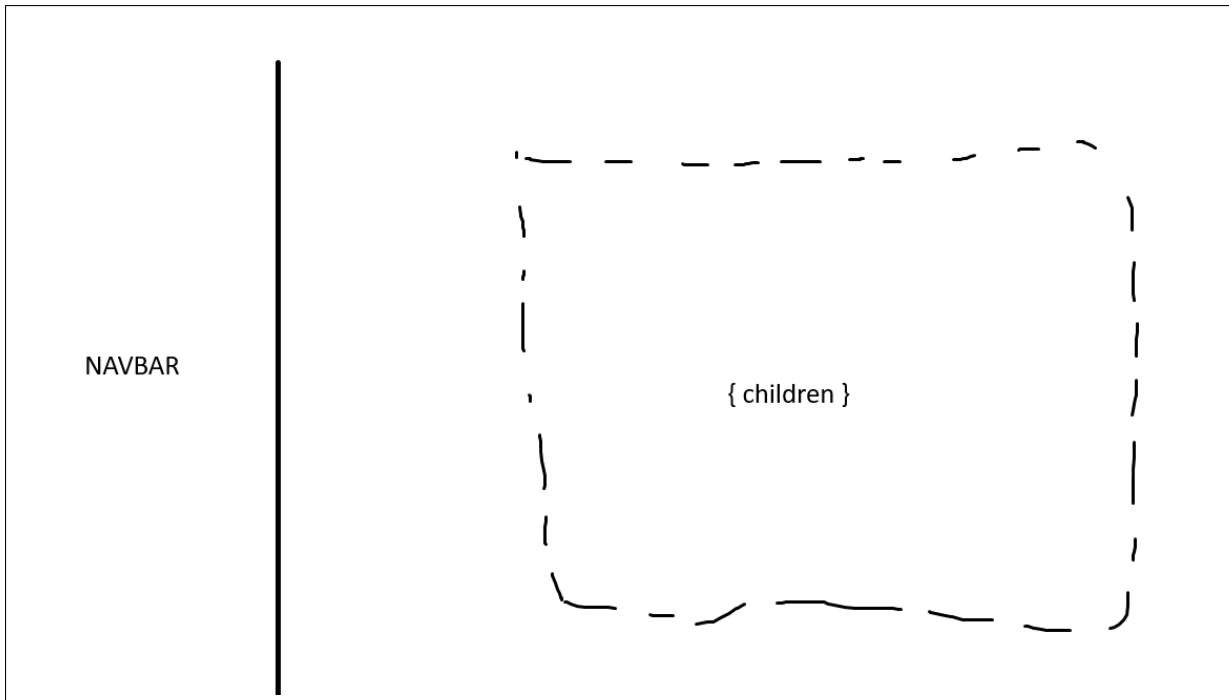
1 import './ui/globals.css';
2 import { Inter } from 'next/font/google';
3 import Navbar_client from './ui/navbar/navbar-client'
4 import Navbar_server from './ui/navbar/navbar-server'
5
6 const inter = Inter({ subsets: ['latin'] })
7
8 export default function Layout({ children }) {
9
10    return (
11        <html lang="en">
12            <body className={`${inter.className} antialiased flex
13                justify-start'>
14                <Navbar_client>
15                    <Navbar_server />
16                </Navbar_client>
17                {children}
18            </body>
19        </html>
20    );
}

```

Figur 11: Viser hvor layout.jsx (markeret rød) befinder sig i projektets filstruktur samt dens inhold

**Layout.jsx-filen** Den importerer navbaren på klient- og serverside, hvorefter denne renderes, hvilket resulterer i, at den ikke behøves genrenderes hele websittets livscyklus.

Måden, hvorpå navbaren forbliver det samme sted på div. webpages, er ved, at layout-filen agerer en overordnet skabelon, hvorefter subdomænernes og forsidsens pagefils kode tilføjes under pladholderen children. Se figur (12) nedenfor for forståelse.



Figur 12: Viser, hvordan layoutfilen påvirker webpagesnes rendering.

**page.jsx-filen** Page.jsx-filen er hjemmesidens forsidefil; det første forbruger møder. Page.jsx-filen indeholder tre primære HTML-sektioner (`<div>`), der på brugeroverfladen fremstår som kasser; som igen er stilet med CSS-tailwind (se paragraf 8.1.2).

**Skemakoden** Skemakoden er den del af Lectio-applikationen, vi har brugt mest tid på, da denne også dag-til-dag er vigtigst samt involverer databaseinfrastruktur.

Koden kan nedbrydes til dage, der kan nedbrydes til moduler. Dagene består et array (liste), hvor hvert et element i arrayet (listen) er et kodeobjekt, der indeholder modulsinformation, herunder faget, underviseren, lokation (lokale), noter, lektier samt metadata som synlighed (om folk skal se dig). Envidere indeholder det også et autogeneret ID, der anvendes til autogeneret web pages, der viser detaljered information om det individuelle modul.

```

1 // Monday
2 const scheduleMon = [
3     { subject: 'MA', teacher: 'jacs', room: 'SLWI-4110', id: '1fxh',
4       visibility: 'show', note: 'Plan: Potensfunktioner, eksponentiel
      vaekst', homework: '03_opg_polyomier.ipynb' },
5
6     { subject: 'MA', teacher: 'jacs', room: 'SLWI-4110', id: '123afa',
7       visibility: 'show', note: '', homework: '' },
8
9     { subject: 'Fy', teacher: 'hnpo', room: 'SLWI-4110', id: 'qw123ac',
10       visibility: 'show', note: 'Ingen lektier. Vi ser paa hvad vi skal i
      timen. Forberedelse til S02 i naeste uge. we we jwnkjkjn
      kjnkjnjn', homework: '' },
11
12    { subject: 'Fy', teacher: 'hnpo', room: 'SLWI-4110', id: 'xcv12',
13      visibility: 'show', note: '', homework: '' },
14
15    { subject: 'sa', teacher: 'mmje', room: 'SLWI-4110', id: 'sdvf23',
16      visibility: 'show', note: 'Det senmoderne samfunds kendeteogn',
      homework: '' },
17
18    { subject: 'So', teacher: 'kibo', room: 'SLWI-4110', id: 'sdf123',
19      visibility: 'show', note: '', homework: 'Hold øje med, hvordan du
      tager noter i de forskellige fag.' },
20
21    { subject: '', teacher: '', room: '', id: 'sdv135', visibility: 'hide',
      note: '', homework: '' },
22 ];

```

Figur 13: Viser et eksempel koden, der skal til for en dag.

Alle dagene har et tilsvarende array med objekter. Selve skemaet er en kodetabel, der består af rækker (tablerowelement). SkemaModule-komponentet tager imod argumenter, hvilke hentes fra den tidligere sete kode.

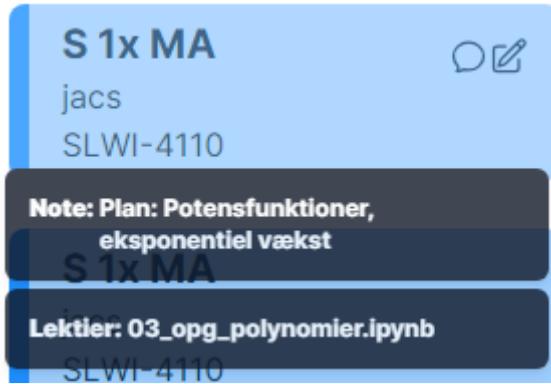
```

1 <td><SkemaModule subject={scheduleMon[0].subject}
2 <td><SkemaModule subject={scheduleTue[0].subject}
3 <td><SkemaModule subject={scheduleWed[0].subject}
4 <td><SkemaModule subject={scheduleThu[0].subject}
5 <td><SkemaModule subject={scheduleFri[0].subject}
```

Figur 14: Viser at »SkemaModule«-komponentet tager første (nulte i kode) objekt i schedule<dag>'s kodeproperty (properties er en værdi, der tilhører et objekt). SkemaModule modtager langt flere properties fra de respektive objekter, fx lærer, fag osv.; det er et udsnit–se appendiks E for resterende kode.

**SkemaModule.jsx** SkemaModule.jsx-filen indeholder det komponent, der referes tidligere »SkemaModule«, heri bliver argumenterne indsatt i HTML-kode, hvor der et script, der tjekker om args.note er tom, altså om der er lektionnoter, for det pågældende modul–hvis denne ikke er tom (altså *indeholder*), viser den et ikon, og det samme sker med args.homework, således at "lektiekonet" vises. Ligeledes dukker "tooltips" også op, når bru-

gerens mus er over et modulelement, dog uddover at der tjekkes om argumenterne/propertiesne er tomme, importeres værdien og renderes i selve tooltippet. Se figur for forståelse:



Figur 15: Viser et modul, hvis args.note og args.homework-argumenter/properties ikke er tomme samt har værdier; ligeledes de respektive tooltips og ikoner.

**Datohåndteringssystem** Datahåndteringssystemet fungerer på den måde, at der op i toppen defineres et array med datoer, dette er datoerne, der fremgår på vores skemaeksempel; hvis softwaren skulle distribueres ville disse blive indhentet via en API (Application programming interface), således datoerne bliver dynamisk opdateret.

Forneden fra linje 4-10 indhentes dagens dato, hvorefter denne anvendes til at skrive dags dato. Herefter laves en konstant, der tager variablerne »date« og »month«, disse anvendes senere ned til at blive sammenlignet med »Current

```

1 const weekNumbers = ["29/4", "30/4", "1/5", "2/5", "3/5"];
2
3
4 const today = new Date();
5 const month = today.getMonth() + 1;
6 const year = today.getFullYear();
7 const date = today.getDate();
8 const currentDate = month + "/" + date + "/" + year;
9
10 const currentFormattedDate = month + "-" + date + "-" + year;
11
12 const weekDate = date + "/" + month;
13
14 var currentWeekNumber = require('current-week-number');
15
16 // Her er en masse kode imellem se appendiks F
17
18 { () => {
19     if (weekNumbers[0] == weekDate) {
20         return (
21             <span className="px-10 p-4 rounded-xl bg-[#1E90FF] text-white font-normal">Mandag <strong>{ weekNumbers[0] }</strong></span>
22         );
23     } else {
24         return (
25             <span className="px-10 p-4 rounded-xl bg-slate-200 font-normal">Mandag <strong>{ weekNumbers[0] }</strong></span>

```

```

26      );
27  }
28
29 })()

```

Herefter tjekker koden om den pågældende ugedag er lig med dags dato, hvorefter dennes respektive UI-element i så fald gøres blåt for at indikere til brugeren, at denne dag er dags dato.

**Dynamic Routes (slugs)** Dynamic routes er "ruter" til data, hvis specifikke lokation vi ikke kender i forvejen (genererer subdomæner dynamisk ud fra input). Dynamic routes anvendes i dette tilfælde til automatisk generation af sider med et respektivt moduls deltajeret information og oversigter, herunder vhaæftelser. Denne genereres ved, at der i skemasubdomænet eksisterer et subsubdomæne med square brackets herom [], hvilket fortæller dokumentet er der er tale om et dynamisk sub(sub)domæne, hvori der ligger en page.jsx-fil tilsvarende til dem, vi har gennemgået tidligere, dog er denne 100% dynamisk, og der specificeres, hvordan information skal formateres, men selve informationen / indholdet hentes via et API fra en anden webserver / serverdatabase. Datasvaret, vi ville få, er samme type som vores skemakoder anvender (objekter), ergo næsten ens, da dette også er .JSON-format—se tidl. figur for dataeksempel [13](#).

Uge 20 - 2024					
	Mandag 29/4	Tirsdag 30/4	Onsdag 1/5	Torsdag 2/5	Fredag 3/5
1.	S 1x MA jacs SLWI-4110	S 1x En2 mgja SLWI-4110	S 1x Ke pgra SLWI-4110	S 1x Ti hypo SLWI-4110	
2.	S 1x MA jacs SLWI-4110	S 1x En2 mgja SLWI-4110	S 1x Ke pgra SLWI-4110	S 1x Ti hypo SLWI-4110	S 1x En2 mgja SLWI-4110
3.	S 1x Fy hypo SLWI-4110	S 1x bi bhjo SLWI-4110	S 1x MA jacs SLWI-4110	S 1x DA kibo SLWI-4110	S 1x kit retv SLWI-4110
4.	S 1x Fy hypo SLWI-4110	S 1x bi bhjo SLWI-4110	S 1x MA jacs SLWI-4110	S 1x DA kibo SLWI-4110	S 1x kit retv SLWI-4110
5.	S 1x sa mmje SLWI-4110	S 1x sa mmje SLWI-4110	S 1x Ti hypo SLWI-4110		S 1x Fy hypo SLWI-4110
6.	S 1x So kibo SLWI-4110		S 1x Ti hypo SLWI-4110		
7.					

Figur 16: Viser skemaet.

## 8.2 Bookingsystem

Vi har valgt at fjerne bookingsystems-delen for dette produkt; dog er den skidegod ide, og det var blevet en del af produktet. Det er af rent pragmatiske årsager, at vi vælger ikke at udvikle systemet, det havde garanteret været sjovt. Skitserne over dette fremgår af appendiks [C](#).

# 9 Produktionsforberedelse

## 9.1 Website-hosting

Der er to muligheder for hosting af websittet. Der er fjernhosting, hvor man lejer sig ind hos en central webserver, fx Vercel. Der er den anden mulighed, local hosting, hvor vi opstiller en decentral server på fx en skoles grund, hvor man så selv indkøber serverudstyr, veligeholder dette in-house eller igennem indkaldte teknikere...

Det er gratis at anvende services som Vercel indtil et vist serverload opnås, hvorefter det koster penge. Jf. deres hjemmeside<sup>[3]</sup> kan hjemmesiden kun tigås en million gange per måned ved gratis abonnement, ti millioner ved 20\$ per måned, men da vi forventer, at servicere et konservativt estimat på 50.000 gymnasieelever (baseret på Danmarks statistik [4]) om dagen, vil dette hurtigt blive overskredet, ergo er der tale om en unik dataaftale, hvorfor det nok bliver nok bliver markant dyre, men vil garanteret være billigere i sidste ende.

## 9.2 RFID-løsning / låsemekanismesystem

### 9.2.1 Fremtidig Plan

Da vi startede med projektet havde jeg meget høje forventninger til hvor meget jeg kunne nå at lave, selvom de ikke blev en realitet, har jeg stadigvæk tænkt mig at dele mine ideer.

Vores oprindelige ide var at bygge hele systemet, både med dørlukker, RFID scanner, og et komplet software system med database der ville havde givet elever og lærere mulighed for at booke lokaler og se hvem der var tilstede. Vores plan for at bygge døren var at købe en billig branddør, vi ville derefter have opsæt en "skelet" dør (en dør som ikke sider fast i en væg), så vi ville have haft mulighed for at fremvise, hvordan produktet ville virke i praksis. Vi havde tænkt os at bestille en elektronisk dørlukker, som vi ville have koblet op til vores ESP32 i vores RFID kasse, vores plan var at trække 12v (Ledninger med 12 volts spænding) for at undgå højstrøms lovgivningen da 12v ledninger ikke indgår under denne lovgivning, dette ville have signifikant sænket omkostninger for opsætning af systemet på skolen. Vi havde tænkt os at benytte et relæ (er et mekanisk kontakt), til at styre både dørlukkeren og den lås som vi ville have sat i dørkarmen for at undgå kompliceret installation i dørene, også i forhold til tilfælde, hvor det kunne blive nødvendigt at udskifte døre. Dette ville også gøre designet mere simpelt og nemmere for os at konstruere. Desværre nåede vi aldrig så langt med vores planer, dette skyldes en del grunde men jeg har tænkt mig at nævne et par af de lidt større fejl og udfordringer der var skyld i at det ikke blev en realitet.

1. **For store ambitioner** I starten af projektet havde vi en masse ideer som vi gerne ville føre til livs, dette resulterede i at størstedelen af vores tid blev brugt på at finde produkter og undersøge installations metoder ud fra vores meget store ambitioner, dette er jo nødvendigvis ikke en dårlig ting, men det var skyld i at vi i det første stykke tid ikke kom i gang med den indledende konstruktion da vi var fokuseret på alle de mange ting vi ville. Da det så endeligt gik op for os at vi var ved at løbe tør for tid, havde vi allerede brugt en signifikant mængde af hele projekttiden.
2. **For lidt prototyping** Vi brugte rigtigt meget tid på at undersøge og prøve at udvikle et perfekt produkt før vi vidste om det overhovedet ville virke i praksis, hvis

vi havde fokuseret mere på at lave nogen prototyper og eksperimentere i starten havde vi realiseret nogen af de design fejl der senere blev åbenlyse da vi lavede vores første prototype. Dette resulterede i at da vi endelig gik i gang med at konstruere og teste vores første prototype var vi nød til at gå tilbage til tegnebrættet og retænke vores design, hvilket resulterede i at der var mange ting vi ikke nåede i mål med da vi endeligt havde et fungerende design.

### **9.2.2 Produktion**

bla bla

## **9.3 Teknologianalyse**

Produktionen og veligholdesen af Lectio-rework applikation skal udliciteres til softwareudviklere fra Indien. Som udgangspunkt skal hovedudviklingen af Lectio-reworket forgå i Danmark. I Danmark er infrastrukturen helt fin, og den er god nok i Indien til vores formål. Kulturen i Indien er meget inbydende til softwarevirksomheder som denne og arbejder gerne for et sådan projekt for den rette løn. Import / eksport til Kina ift. RFID-systemets produktion er særligt nemt, da logistikinfrastrukturen er veletableret og -udbygget. Software kan let eksporteres fra Indien.

# **10 Evaluering, vurdering og konklusion**

## **10.1 Test af produkt**

Vi har testet produktet undervejs ved at udsættet testpersoner for produktet, hvorefter de har vurderet produktets æstetik og brugervenlighed rent kvalitativt, hvilket vi har taget til og ageret på.

## **10.2 Teknologivurdering**

### **10.2.1 Konsekvensvurdering**

Vi har lavet en konsekvensvurdering af Lectio, hvor vi er kommet frem til, at det er en ineffektiv IT-løsning, hvorefter vi har lavet et rework af dette med disse fejl og mangler rettet. Om hvorvidt vores applikationspakke bedre respekterer sine brugers tid, dvs. sparer den, vil kræve en kvantiativ undersøgelse, der først kan igangsættes, når produktet er blevet frigivet til flere mennesker. Envidere ville en kvalitativ undersøgelse skulle igangsættes for at vurdere om brugervenligheden er forbedret.

## **10.3 Procesevaluering**

Vi har anvendt Gantt-diagram, skitser og systematiske metoder til at finde et fyldesgørende produkt til et relevant samfundsmæssigt problem, der vedrører vores studiemiljø. Det er gået udemærket, dog har vi skulle hugge en hæl og klippe en tå for at nå i mål–det er overordnet gået meget godt med den tid og de ressourcer, vi har haft til rådighed, dog var projektet en smule for ambitiøst. En anden gang vil vi anvende et andet markup-sprog til at holde styr og dokumentere vores projekt i, nemlig Org-Mode fremfor L<sup>A</sup>T<sub>E</sub>X, da dette vil gøre, at vi bedre kan integrere kode i projektet samt fokusere mere på andre aspekter af projekter samt tillader os at anvende den overlegne Emacs til sit fulde potentiale.

## **10.4 Konklusion**

All-in-all et vellykket projekt.

## Bibliografi

1. A/S, M. *Om Lectio* <https://www.lectio.dk/pdf/omlectio.pdf> (2024).
2. Next.js. *Next.js Project Structure* <https://nextjs.org/docs/getting-started/project-structure> (2024).
3. Vercel. *Pricing* [vercel.com/pricing](https://vercel.com/pricing) (2024).
4. statistik, D. *Ungdomsuddannelser* <https://www.dst.dk/da/Statistik/emner/uddannelse-og-forskning/fuldtidsuddannelser/ungdomsuddannelser> (2024).

## Appendiks

## A Projektbeskrivelse

# Projektbeskrivelse

Jeppe, Alexander, Andreas, David

## 1 Problemanalyse - intro

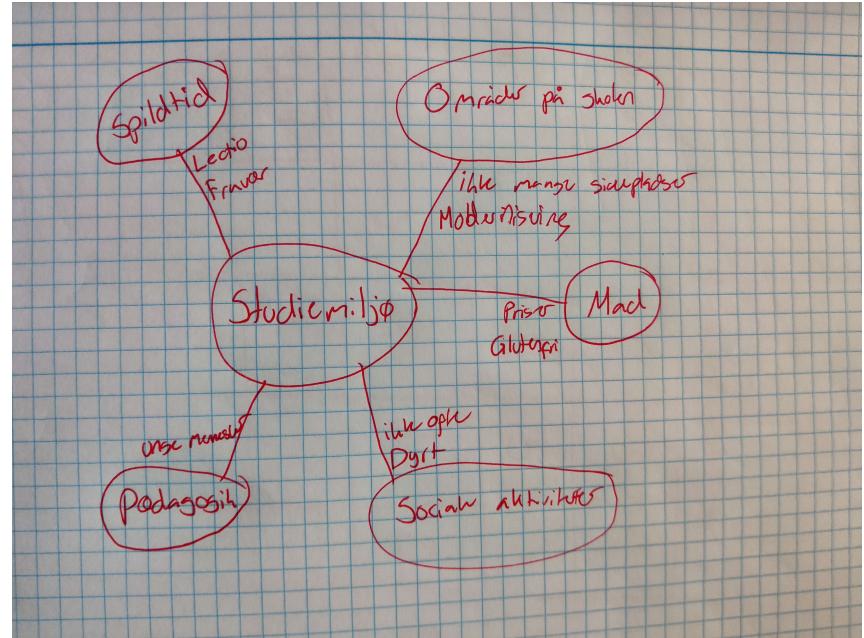
- Samfundsmaessigt
- Brandsikkerhed
- Fraværsregistering
- modernisering af IT Systemer og forbedret infrastruktur.

## 2 Problemidentifikation

### 2.1 Idegenerering

#### 2.1.1 Mindmap

Vi har anvendt et mindmap til at danne et overblik over, hvilke problemer vi i dette projekt kan bearbejde.



Figur 1: Viser det anvende mindmap.

#### 2.1.2 Lyskurven

Efter vi lavede Mindmappet, valgte vi at benytte os af lyskurve-modellen til at sorterer ivores ideer. Lyskurven er et godt værktøj, som man bruger til at sortere i ens problemstillinger. Lyskurven fungerer ved, at man opstiller tre kategorier for at inddelte ens ideer efter, hvor (tænkt) brugbare ideerne er. Så, man tildeler de problemer, man i gruppen gerne vil arbejde videre med og ser potentielle i, i den grønne kategori. Så har man også en gul kategori, som man bruger til ideer, man ikke er helt sikker på kan lykkedes, men der er i en backup-fase, så de kan tages i brug, hvis ens grønne ideer går i vasken. Til sidst har man den røde kategori, her befinner de ideer, man i gruppen har valgt at kasserer. Vores Lyskurve så således ud:

Spildtid, områder på skole	Grøn
Pædagogik	Gul
Sociale aktiviteter, mad	Rød

Tabel 1: Viser et meget abstrakt lyskurvediagram i form af en tabel; det vi anvendte

## 2.2 Identificering af nøgleproblem

I vores lyskurvediagram (se tabellen 1) har vi to i den grønne, gode, kategori, ergo skal vi endvidere vælge en af disse at arbejde med. Vi har anvendt følgende spørgsmål til at

indsnævre valget:

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?  
\*\* Spørgsmålene er hentet fra ”projektarbejdet-bogen via Systime <sup>1</sup>

Således har vi besvaret spørgsmålene:

- 1) Tidspild er interessant, fordi det særligt vedrører 1.x-klassen, da det er en meget stor klasse med tilsvarende få lærer, ergo er der på forhånd en dårlig tidsallokering per elev per modul samt er der mange ting, der gøres på ineffektive muligheder; ergo er casen oplagt. Områder på skolen er også interessant, da det berører os alle, men det er ikke en videre spændende problemstilling at fordybe i.
- 2) Redundans - jf. besvarelserne ovenover
- 3) Ja, bl.a. da vi er kede af, hvor meget tid vi spilder. De fysiske områder kunne godt trænge til lidt modernisering.
- 4) Produktet med er til gavn for samtlige af det danske kongeriges studerende og deres respektive undervisere. Forbedring af de fysiske områder ville først og fremmest komme os til gode.



Ergo vælger vi at arbejde videre med tidspildsproblematikken, da denne er mest interessant baseret på svarene.

### 3 Produktudkast

Vores Lectio Renovering modernisser HTX' IT-systemer. Det vil vi blandt andet gøre ved at lancere et identitesbaseret system på RFID, hvilket gør en ende på spildt tid med fraværstagning og eventuelt falsk fravær, da dette automatisk registreres, når du scanner dit ID-kort.

#### 3.1 Lokale-Booking-System

Vores Lokale-Booking-System vil revolutionere måden, hvorpå vi booker lokaler. Det vil gøre en ende på unødig tidsspild, altså den tid, det tager at gå op på kontoret, frem og tilbage med nøgler.

#### 3.2 Smartdøre

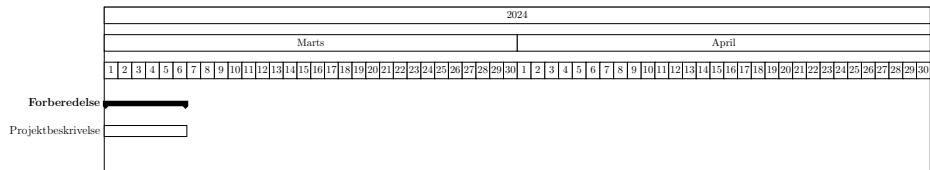
Vores Smartdøre vil forbedre brandsikkerheden og fungerer sammen med booking-systemet og elevernes identifikationskort. I tilfælde af brand vil alle skolens døre låse op og lukke automatisk så alle rum bliver isoleret, og brænderen bliver forsøgt kvalt.

### 3.3 Lectio Renovering

Det vil også sige, at hvis et lokale nu er låst, og uheldet er ude, hvor man skal flygte ud af et vindue, der er bag et aflåst lokale, vil man nu kunne spare patientelt tabte liv.

<sup>1</sup><https://projektarbejdet.systime.dk/?id=145>

## 4 Tidsplan



Figur 2: Viser Gantt-Diagram over vores foreløbige tidsplan

## B Logbog

# Logbog

Jeppe, Alexander, Andreas, David

## Indhold

1	01/02-2024	2
2	26/02-2024	2
3	06/03-2024	2
4	14/03-2024	2
5	15/03-2024	2
6	16/03-2024	2
7	17/03-2024	2
8	10/04-2024	3
9	11/04-2024	3
10	18/04-2024	3

B3  
1

## 1 01/02-2024

- Opsat L<sup>A</sup>T<sub>E</sub>Xdokument.
- Formateret dokument.
- Valgt deloplæg.

## 2 26/02-2024

- Projektbeskrivelse.
- Med videre beslutning om tilføjelse om flugtplan på produktet.

## 3 06/03-2024

- Uploade dokument til teams.

## 4 14/03-2024

- Vi har gennemgået logbogsteknik, brainstorm om brænddører og brandsikkerhed. Dertil arbejdet på problemanalyse–tilføjet tidsspild ift. brand og andet.
- Opsatte Lectio renoveringsprojekt via ext.js.

## 5 15/03-2024

- Lavet skemagrid.
- Gjorde således, at skemabrikindsætnigen kunne oplisteres vertikalt fremfor horizontalt altså per dagsbasis, hvilket demonstrerer dynamisk skemanedhentning fra fx database.

## 6 16/03-2024

- Justerede tidsplan, således at Gantt-diagrammet kan sættes ved hjælp af datoværdier fremfor heltalsværdier.

## 7 17/03-2024

- Forbedret lectio renoveringprojekts design.
  - Tilføjet hover-effekt på skemabrikker.
  - Tilføjet dynamisk-opdaterende ugesystem.
  - Tilføjet dynamisk-opdaterende lektie-note-ikonssystem per skemabrik.

- Forbedret appendiks-system, således at disse fremgår mere klart samt tilføjet paginering til disse for nemme orientering.
- Tilføjet kapitler til projektudvikling.
- Forbedret tidsplanskalering (størrelse på tekst).
- Tilføjet præliminærside „Indledning“, hvori tekstopstætnings- og mere generelle overvejelser vil fremgå.
- Ændrede paginering afhængigt af denne er præliminær (romertal-iii), en del af appendiksen (Appendiks bogstav samt arabisk tal-A1 eller A10) eller en del af hovedværket (arabisk tal-1-10).

## 8 10/04-2024

- Vi har oprettet og lavet en forbedret to-dolist,
- Vi er nødsaget til at tage et vigtigt valg og desværre kommet frem til, at vi bliver nødt til at kassere booking-system-ideen.
- Vi har tilføjet iconer

## 9 11/04-2024

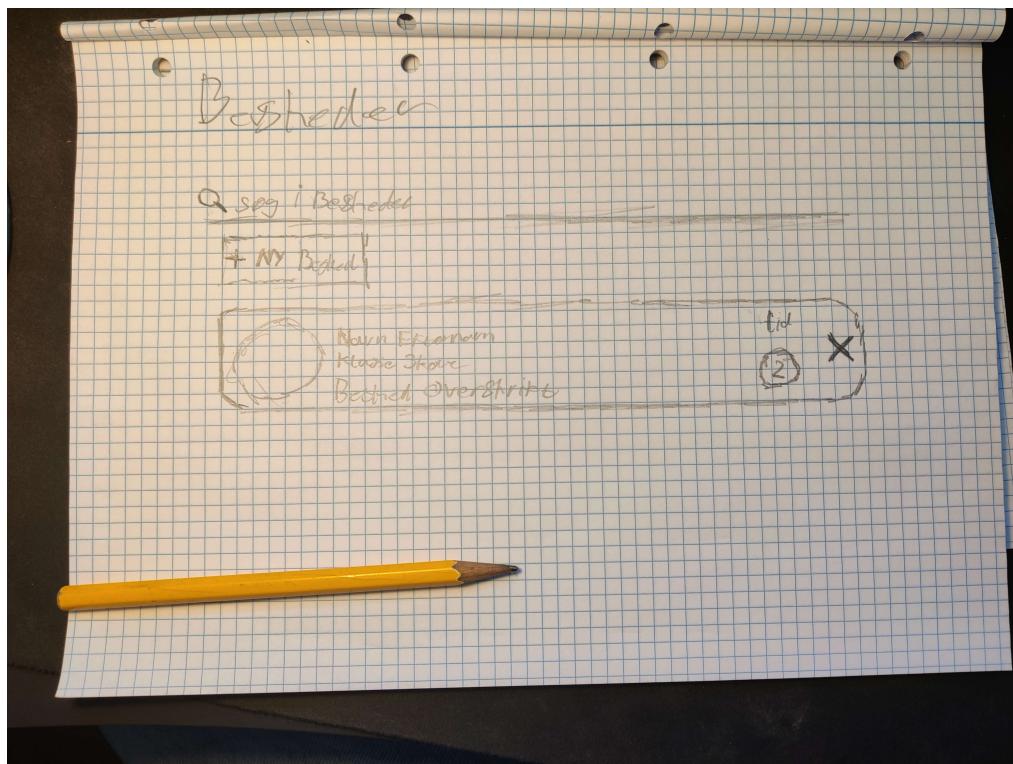
B32

- idag har vi korrigeret vores tidsplan, og dertil forbedret vores overblik
- Vi har kommet i besiddelse af Flugtvejsplan, og vi har planer om at tilføje den til produktet
- Yderligere har vi brainstormet idéer til udseende på produktet, booking af lokaler m.m

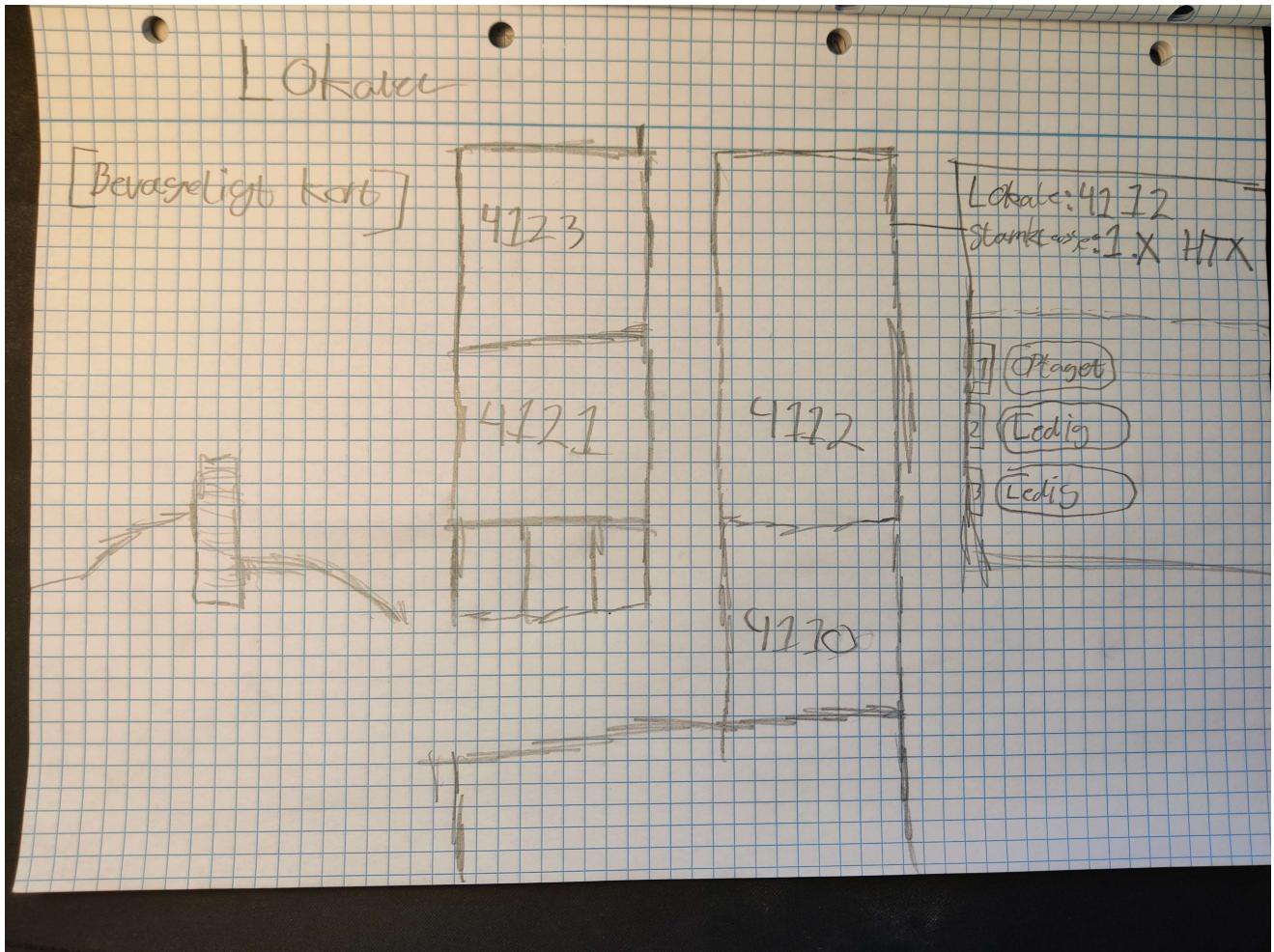
## 10 18/04-2024

- Vi har idag været til gruppe samtale med henrik vedrørende vores projekt, vi er godt på vej og har gode idéer og tanker til hvordan vores projekt kan/komme til at se ud.
- Vi har uploadet dokument til at teams, der omhandler vores projekt
- vi har en masse Positive tanker omkring det her, vi tror det kan blive rigtigt godt hvis det bliver gjort på den rigtige måde og det gør det

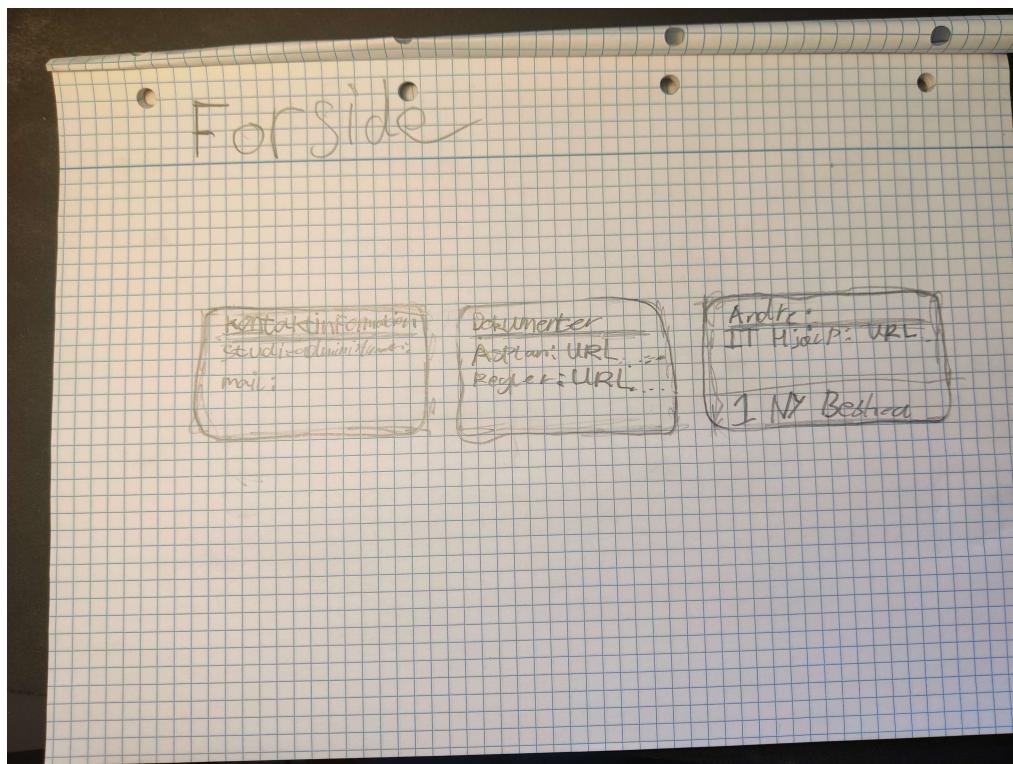
## C Arbejdsskitser



Figur 17: Viser skitsen for lectiobesked-featuren



Figur 18: Viser skitsen for lectiolokalebooking-featuren



Figur 19: Viser skitsen for lectioskema-featurene

## D Modulbrikkoden

```

1 import './skema.css';
2 import { ChatBubbleOvalLeftIcon } from "@heroicons/react/24/outline";
3 import { PencilSquareIcon } from "@heroicons/react/24/outline";
4 import Link from 'next/link'
5
6
7 export default function skemaModule( args ) {
8
9     if ( args.visibility == 'hide' ) {
10         return (
11             <div className="min-h-[88px]"></div>
12         );
13     } else {
14         return (
15             <Link href={`/skema/${args.id}`}>
16                 <div className="bg-[#9CCEFF] border-l-[10px] border-l-[#1E90FF]
17                     p-4 rounded-lg max-w-[275px] max-h-[88px] flex justify-
18                     between cursor-pointer hover:opacity-80 group">
19                     <div className="flex flex-col text-[#0D3F70] justify-center"
20                         >
21                         <span className="font-bold text-xl">S 1x { args.subject
22                             }</span>
23                         <span>{ args.teacher }</span>
24                         <span>{ args.room }</span>
25                     </div>
26
27                     {/* Tooltip */}
28                     <div className='absolute'
29                         <div className='max-w-[275px] flex flex-col flex-wrap mt-16'
30                             >
31                         <div>
32                             {(() => {
33                                 if ( args.note != '' ) {
34                                     return (
35                                         <span className='sidebar-tooltip group-hover
36                                             :scale-100 relative left-[-30px]'><h1
37                                             className='font-black'>Note:</h1>&nbsp;<
38                                             span>{ args.note }</span></span>
39                                         );
40                                 } else {
41                                     return (
42                                         '',
43                                         );
44                                 }
45                             })()
46                         </div>
47                         <div>
48                             {(() => {
49                                 if ( args.homework != '' ) {
50                                     return (
51                                         <span className='sidebar-tooltipH group-
52                                             hover:scale-100 relative left-[-30px]'><
53                                             h1 className='font-black'>Lektier:</h1>&
54                                             nbsp;<span>{ args.homework }</span></
55                                             span>
56                                     );
57                                 }
58                             })()
59                         </div>
60                     </div>
61                 </div>
62             </Link>
63         );
64     }
65 }

```

```

47         } else {
48             return (
49                 '',
50             );
51         }
52     })()
53     </div>
54 </div>
55 </div>
56 <div className="text-[#0D3F70] flex flex-row float-right">
57     <div>
58         { () => {
59             if ( args.note != '' ) {
60                 return (
61                     <span><ChatBubbleOvalLeftIcon className=
62                         "h-5"/></span>
63                 );
64             } else {
65                 return (
66                     <span></span>
67                 );
68             }
69         })()
70     </div>
71     <div>
72         { () => {
73             if ( args.homework != '' ) {
74                 return (
75                     <span><PencilSquareIcon className="h-5"
76                         /></span>
77                 );
78             } else {
79                 return (
80                     <span></span>
81                 );
82             }
83         })()
84     </div>
85     </div>
86     </div>
87     </Link>
88     );
89 }
90 }

```

## E Skemakomponentkode

```

1 <tbody>
2
3 /* Module 1 */
4 <tr>
5   <td> <span className="h-[50px] bg-slate-300 py-10 px-2 rounded-e-md
6     text-slate-600 font-bold">1.</span> </td>
7   <td><SkemaModule subject={scheduleMon[0].subject} teacher={scheduleMon[0].teacher} room={scheduleMon[0].room} visibility={scheduleMon[0].visibility} note={scheduleMon[0].note} homework={scheduleMon[0].homework} id={scheduleMon[0].id}/></td>
8   <td><SkemaModule subject={scheduleTue[0].subject} teacher={scheduleTue[0].teacher} room={scheduleMon[0].room} visibility={scheduleTue[0].visibility} note={scheduleTue[0].note} homework={scheduleTue[0].homework} id={scheduleTue[0].id}/></td>
9   <td><SkemaModule subject={scheduleWed[0].subject} teacher={scheduleWed[0].teacher} room={scheduleWed[0].room} visibility={scheduleWed[0].visibility} note={scheduleWed[0].note} homework={scheduleWed[0].homework} id={scheduleWed[0].id}/></td>
10  <td><SkemaModule subject={scheduleThu[0].subject} teacher={scheduleThu[0].teacher} room={scheduleThu[0].room} visibility={scheduleThu[0].visibility} note={scheduleThu[0].note} homework={scheduleThu[0].homework} id={scheduleThu[0].id}/></td>
11 <td><SkemaModule subject={scheduleFri[0].subject} teacher={scheduleFri[0].teacher} room={scheduleFri[0].room} visibility={scheduleFri[0].visibility} note={scheduleFri[0].note} homework={scheduleFri[0].homework} id={scheduleFri[0].id}/></td>
12 </tr>
13
14 /* Module 2 */
15 <tr className="m-2">
16   <td> <span className="h-[50px] bg-slate-300 py-10 px-2 rounded-e-md
17     text-slate-600 font-bold">2.</span> </td>
18   <td><SkemaModule subject={scheduleMon[1].subject} teacher={scheduleMon[1].teacher} room={scheduleMon[1].room} visibility={scheduleMon[1].visibility} note={scheduleMon[1].note} homework={scheduleMon[1].homework} id={scheduleMon[1].id} /></td>
19   <td><SkemaModule subject={scheduleTue[1].subject} teacher={scheduleTue[1].teacher} room={scheduleTue[1].room} visibility={scheduleTue[1].visibility} note={scheduleTue[1].note} homework={scheduleTue[1].homework} id={scheduleTue[1].id} /></td>
20   <td><SkemaModule subject={scheduleWed[1].subject} teacher={scheduleWed[1].teacher} room={scheduleWed[1].room} visibility={scheduleWed[1].visibility} note={scheduleWed[1].note} homework={scheduleWed[1].homework} id={scheduleWed[1].id} /></td>
21   <td><SkemaModule subject={scheduleThu[1].subject} teacher={scheduleThu[1].teacher} room={scheduleThu[1].room} visibility={scheduleThu[1].visibility} note={scheduleThu[1].note} homework={scheduleThu[1].homework} id={scheduleThu[1].id} /></td>
22   <td><SkemaModule subject={scheduleFri[1].subject} teacher={scheduleFri[1].teacher} room={scheduleFri[1].room} visibility={scheduleFri[1].visibility} note={scheduleFri[1].note} homework={scheduleFri[1].homework} id={scheduleFri[1].id} /></td>
23 </tr>
24
25 /* Module 3 */
<tr>
  <td> <span className="h-[50px] bg-slate-300 py-10 px-2 rounded-e-md text
    -slate-600 font-bold">3.</span> </td>

```

```

26 <td><SkemaModule subject={scheduleMon[2].subject} teacher={  
27   scheduleMon[2].teacher} room={scheduleMon[2].room} visibility={  
28     scheduleMon[2].visibility} note={scheduleMon[2].note} homework={  
29       scheduleMon[2].homework} id={scheduleMon[2].id} /></td>  
30 <td><SkemaModule subject={scheduleTue[2].subject} teacher={  
31   scheduleTue[2].teacher} room={scheduleTue[2].room} visibility={  
32     scheduleTue[2].visibility} note={scheduleTue[2].note} homework={  
33       scheduleTue[2].homework} id={scheduleTue[2].id} /></td>  
34 <td><SkemaModule subject={scheduleWed[2].subject} teacher={  
35   scheduleWed[2].teacher} room={scheduleWed[2].room} visibility={  
36     scheduleWed[2].visibility} note={scheduleWed[2].note} homework={  
37       scheduleWed[2].homework} id={scheduleWed[2].id} /></td>  
38 <td><SkemaModule subject={scheduleThu[2].subject} teacher={  
39   scheduleThu[2].teacher} room={scheduleThu[2].room} visibility={  
40     scheduleThu[2].visibility} note={scheduleThu[2].note} homework={  
41       scheduleThu[2].homework} id={scheduleThu[2].id} /></td>  
42 <td><SkemaModule subject={scheduleFri[2].subject} teacher={  
43   scheduleFri[2].teacher} room={scheduleFri[2].room} visibility={  
44     scheduleFri[2].visibility} note={scheduleFri[2].note} homework={  
45       scheduleFri[2].homework} id={scheduleFri[2].id} /></td>  
46 </tr>  
47 /* Module 4 */  
<tr>  
48 <td> <span className="h-[50px] bg-slate-300 py-10 px-2 rounded-e-md text  
49 -slate-600 font-bold">4.</span> </td>  
50 <td><SkemaModule subject={scheduleMon[3].subject} teacher={  
51   scheduleMon[3].teacher} room={scheduleMon[3].room} visibility={  
52     scheduleMon[3].visibility} note={scheduleMon[3].note} homework={  
53       scheduleMon[3].homework} id={scheduleMon[3].id} /></td>  
54 <td><SkemaModule subject={scheduleTue[3].subject} teacher={  
55   scheduleTue[3].teacher} room={scheduleTue[3].room} visibility={  
56     scheduleTue[3].visibility} note={scheduleTue[3].note} homework={  
57       scheduleTue[3].homework} id={scheduleTue[3].id} /></td>  
58 <td><SkemaModule subject={scheduleWed[3].subject} teacher={  
59   scheduleWed[3].teacher} room={scheduleWed[3].room} visibility={  
60     scheduleWed[3].visibility} note={scheduleWed[3].note} homework={  
61       scheduleWed[3].homework} id={scheduleWed[3].id} /></td>  
62 <td><SkemaModule subject={scheduleThu[3].subject} teacher={  
63   scheduleThu[3].teacher} room={scheduleThu[3].room} visibility={  
64     scheduleThu[3].visibility} note={scheduleThu[3].note} homework={  
65       scheduleThu[3].homework} id={scheduleThu[3].id} /></td>  
66 <td><SkemaModule subject={scheduleFri[3].subject} teacher={  
67   scheduleFri[3].teacher} room={scheduleFri[3].room} visibility={  
68     scheduleFri[3].visibility} note={scheduleFri[3].note} homework={  
69       scheduleFri[3].homework} id={scheduleFri[3].id} /></td>  
70 </tr>  
71 /* Module 5 */  
72 <tr>  
73 <td> <span className="h-[50px] bg-slate-300 py-10 px-2 rounded-e-md text  
74 -slate-600 font-bold">5.</span> </td>  
75 <td><SkemaModule subject={scheduleMon[4].subject} teacher={  
76   scheduleMon[4].teacher} room={scheduleMon[4].room} visibility={  
77     scheduleMon[4].visibility} note={scheduleMon[4].note} homework={  
78       scheduleMon[4].homework} id={scheduleMon[4].id} /></td>  
79 <td><SkemaModule subject={scheduleTue[4].subject} teacher={  
80   scheduleTue[4].teacher} room={scheduleTue[4].room} visibility={
```

```

    scheduleTue[4].visibility} note={scheduleTue[4].note} homework={  

    scheduleMon[4].homework} id={scheduleTue[4].id} /></td>  

48 <td><SkemaModule subject={scheduleWed[4].subject} teacher={  

    scheduleWed[4].teacher} room={scheduleWed[4].room} visibility={  

    scheduleWed[4].visibility} note={scheduleWed[4].note} homework={  

    scheduleTue[4].homework} id={scheduleWed[4].id} /></td>  

49 <td><SkemaModule subject={scheduleThu[4].subject} teacher={  

    scheduleThu[4].teacher} room={scheduleThu[4].room} visibility={  

    scheduleThu[4].visibility} note={scheduleThu[4].note} homework={  

    scheduleWed[4].homework} id={scheduleThu[4].id} /></td>  

50 <td><SkemaModule subject={scheduleFri[4].subject} teacher={  

    scheduleFri[4].teacher} room={scheduleFri[4].room} visibility={  

    scheduleFri[4].visibility} note={scheduleFri[4].note} homework={  

    scheduleFri[4].homework} id={scheduleFri[4].id} /></td>  

51 </tr>  

52 /* Module 6 */  

53 <tr>  

54 <td> <span className="h-[50px] bg-slate-300 py-10 px-2 rounded-e-md text  

      -slate-600 font-bold">6.</span> </td>  

55 <td><SkemaModule subject={scheduleMon[5].subject} teacher={  

    scheduleMon[5].teacher} room={scheduleMon[5].room} visibility={  

    scheduleMon[5].visibility} note={scheduleMon[5].note} homework={  

    scheduleMon[5].homework} id={scheduleMon[5].id} /></td>  

56 <td><SkemaModule subject={scheduleTue[5].subject} teacher={  

    scheduleTue[5].teacher} room={scheduleTue[5].room} visibility={  

    scheduleTue[5].visibility} note={scheduleTue[5].note} homework={  

    scheduleTue[5].homework} id={scheduleTue[5].id} /></td>  

57 <td><SkemaModule subject={scheduleWed[5].subject} teacher={  

    scheduleWed[5].teacher} room={scheduleWed[5].room} visibility={  

    scheduleWed[5].visibility} note={scheduleWed[5].note} homework={  

    scheduleWed[5].homework} id={scheduleWed[5].id} /></td>  

58 <td><SkemaModule subject={scheduleThu[5].subject} teacher={  

    scheduleThu[5].teacher} room={scheduleThu[5].room} visibility={  

    scheduleThu[5].visibility} note={scheduleThu[5].note} homework={  

    scheduleThu[5].homework} id={scheduleThu[5].id} /></td>  

59 <td><SkemaModule subject={scheduleFri[5].subject} teacher={  

    scheduleFri[5].teacher} room={scheduleFri[5].room} visibility={  

    scheduleFri[5].visibility} note={scheduleFri[5].note} homework={  

    scheduleFri[5].homework} id={scheduleFri[5].id} /></td>  

60 </tr>  

61 /* Module 7 */  

62 <tr>  

63 <td> <span className="h-[50px] bg-slate-300 py-10 px-2 rounded-e-md text  

      -slate-600 font-bold">7.</span> </td>  

64 <td><SkemaModule subject={scheduleMon[6].subject} teacher={  

    scheduleMon[6].teacher} room={scheduleMon[6].room} visibility={  

    scheduleMon[6].visibility} note={scheduleMon[6].note} homework={  

    scheduleMon[6].homework} id={scheduleMon[6].id} /></td>  

65 <td><SkemaModule subject={scheduleTue[6].subject} teacher={  

    scheduleTue[6].teacher} room={scheduleTue[6].room} visibility={  

    scheduleTue[6].visibility} note={scheduleTue[6].note} homework={  

    scheduleTue[6].homework} id={scheduleTue[6].id} /></td>  

66 <td><SkemaModule subject={scheduleWed[6].subject} teacher={  

    scheduleWed[6].teacher} room={scheduleWed[6].room} visibility={  

    scheduleWed[6].visibility} note={scheduleWed[6].note} homework={  

    scheduleWed[6].homework} id={scheduleWed[6].id} /></td>
```

```
69 <td><SkemaModule subject={scheduleThu[6].subject} teacher={  
    scheduleThu[6].teacher} room={scheduleThu[6].room} visibility={  
    scheduleThu[6].visibility} note={scheduleThu[6].note} homework={  
    scheduleThu[6].homework} id={scheduleThu[6].id} /></td>  
70 <td><SkemaModule subject={scheduleFri[6].subject} teacher={  
    scheduleFri[6].teacher} room={scheduleFri[6].room} visibility={  
    scheduleFri[6].visibility} note={scheduleFri[6].note} homework={  
    scheduleFri[6].homework} id={scheduleFri[6].id} /></td>  
71 </tr>  
72 </tbody>
```

## F Datahåndteringssystem

```

1 export default function skema() {
2 const weekNumbers = ["29/4", "30/4", "1/5", "2/5", "3/5"];
3
4
5 const today = new Date();
6 const month = today.getMonth()+1;
7 const year = today.getFullYear();
8 const date = today.getDate();
9 const currentDate = month + "/" + date + "/" + year;
10
11 const currentFormattedDate = month + "-" + date + "-" + year;
12
13 const weekDate = date + "/" + month;
14
15 var currentWeekNumber = require('current-week-number');
16
17
18 return (
19   <div className="px-10">
20     <Link href={`/ ${scheduleMon[1].subject} - ${currentFormattedDate}`}></Link>
21
22   <div className="container">
23     <div className="w-[80vw] h-[100vh] py-10">
24       <div className="w-full flex justify-center font-bold bg-
25         white py-2">
26         <h1> Uge { currentWeekNumber(currentDate) } - { year
27           }</h1>
28       </div>
29       <table className="w-full h-full bg-white">
30         <thead>
31           <tr className="h-20">
32             <th></th>
33             <th>
34               {(() => {
35                 if (weekNumbers[0] == weekDate) {
36                   return (
37                     <span className="px-10 p-4 rounded-xl bg-[#1E90FF] text-
38                       white font-normal">Mandag <strong>{ weekNumbers[0]
39                         }</strong></span>
40                   );
41                 } else {
42                   return (
43                     <span className="px-10 p-4 rounded-xl bg-slate-200 font-
44                       normal">Mandag <strong>{ weekNumbers[0] }</strong></
45                         span>
46                   );
47                 }
48               })()
49             </th>
50             <th>
51               {(() => {
52                 if (weekNumbers[1] == weekDate) {
53                   return (
54                     <span className="px-10 p-4 rounded-xl bg-[#1E90FF] text-
55                       white font-normal">Tirsdag <strong>{ weekNumbers[1]
56                         }</strong></span>

```

```

50         );
51     } else {
52     return (
53         <span className="px-10 p-4 rounded-xl bg-slate-200 font-
54             normal">Tirsdag <strong>{ weekNumbers[1] }</strong></span>
55     );
56   }) ();
57 </th>
58 <th>
59 { () => {
60   if (weekNumbers[2] == weekDate) {
61     return (
62       <span className="px-10 p-4 rounded-xl bg-[#1E90FF] text-
63           white font-normal">Onsdag <strong>{ weekNumbers[2] }</
64           strong></span>
65     );
66   } else {
67     return (
68       <span className="px-10 p-4 rounded-xl bg-slate-200 font-
69           normal">Onsdag <strong>{ weekNumbers[2] }</strong></span>
70     );
71   }) ();
72 </th>
73 <th>
74 { () => {
75   if (weekNumbers[3] == weekDate) {
76     return (
77       <span className="px-10 p-4 rounded-xl bg-[#1E90FF] text-
78           white font-normal">Torsdag <strong>{ weekNumbers[3] }</
79           strong></span>
80     );
81   } else {
82     return (
83       <span className="px-10 p-4 rounded-xl bg-slate-200 font-
84           normal">Torsdag <strong>{ weekNumbers[3] }</strong></span>
85     );
86   }) ();
87 </th>
88 <th>
89 { () => {
90   if (weekNumbers[4] == weekDate) {
91     return (
92       <span className="px-10 p-4 rounded-xl bg-[#1E90FF] text-
93           white font-normal">Fredag <strong>{ weekNumbers[4] }</
94           strong></span>
95   }) ();

```

```
96          </th>
97      </tr>
98  </thead>
```

## G Eksempel på beskedfunktionalitet

Q søg i beskeder

---

+ Ny besked



Jeppe Bech  
1. X - HTX  
Hey John Doe, jeg kommer ikke i skole imorgen fordi bla

14.22 2 X



Jeppe Bech  
1. X - HTX  
Hey John Doe, jeg kommer ikke i skole imorgen fordi bla

14.22 2 X