



Høgskulen
på Vestlandet

BACHELOROPPGAVE

Måltidsplanlegger – Et verktøy for å
strukturere måltider for barn i
helseomsorg

Dietary planner - A tool for structuring
meals for children in healthcare

Lars Erik Birkeland, Sander Mikael Løtuft og Bernt
Otto von Münchow

Dataingeniør og informasjonsteknologi
Fakultet for ingeniør- og naturvitenskap
Carsten Gunnar Helgesen

22.05.23

Jeg bekrefter at arbeidet er selvstendig utarbeidet, og at referanser/kildehenvisninger til alle kilder som er
brukt i arbeidet er oppgitt, *jf. Forskrift om studium og eksamen ved Høgskulen på Vestlandet, § 12-1.*

TITTELSIDE FOR HOVEDPROSJEKT

Rapportens tittel:	Dato:
Måltidsplanlegger – Et verktøy for å strukturere måltider for barn i helseomsorg	22.05.23
Forfatter(e):	Antall sider u/vedlegg: 45
Lars Erik Birkeland, Sander Mikael Løtuft, og Bernt Otto von Münchow	Antall sider vedlegg: 62
Studieretning:	Antall disketter/CD-er: 0
Dataingeniør, Informasjonsteknologi	
Kontaktperson ved studieretning:	<i>Gradering:</i> Ingen
Carsten Helgesen	
Merknader:	

Oppdragsgiver:	Oppdragsgivers referanse:
Adfectus AS	
Oppdragsgivers kontaktperson:	Telefon:
Christer Kilavik	+47 911 92 227

<p>Sammendrag:</p> <p>Helsevesenet i Norge har store problemer med å ivareta behovene til alle pasientene deres. Mange får ikke den oppfølgingen de trenger, når de trenger det. I flere år har det vært fokus på økt digitalisering for å hjelpe helsearbeidere. Adfectus appen skal bistå med behandlingen av innlagte barn. Den bruker visuell kommunikasjon og spillifisering for at barn skal lettere forstå det som skjer og gi dem muligheten til å påvirke behandlingen deres.</p> <p>I dette prosjektet skal det utvikles en ny modul til Adfectus-appen. Måltidsplanleggeren er interaktivt verktøy for å planlegge kommende måltider. Håpet er at den skal bidra til å gi sunne matvaner, samtidig som at den ikke krever ekstra ressurser for helsepersonell.</p>

Stikkord:

React-Native Redux	Helseomsorg Barn	Mobilapplikasjon
-----------------------	---------------------	------------------

Høgskulen på Vestlandet, Fakultet for ingeniør- og naturvitenskap
 Postadresse: Postboks 7030, 5020 BERGEN Besøksadresse: Inndalsveien 28, Bergen
 Tlf. 55 58 75 00 Fax 55 58 77 90 E-post: post@hvl.no
 Hjemmeside: <http://www.hvl.no>

FORORD

Denne rapporten dokumenter arbeidet med *Måltidsplanlegger*-prosjektet, gjennomført av Bernt Otto von Münchow, Lars Erik Birkeland og Sander Mikael Løtuft, ved Høgskulen på Vestlandet (HVL), våren 2023. Bachelorprosjektet er gjort i samarbeid med Adfectus AS som utvikler en mobilapplikasjon med samme navn.

Gruppen ønsker å takke alle personer ved Eitri Medisinsk Inkubator, og da spesielt Finn-Arne Nielsen fra Retrams AS for hyggelige lunsjsamtaler, Torleif Markussen Lunde og Maija Slaidina fra Eitri for hver eneste vaffelfredag.

Gruppen ønsker å takke alle som har hjulpet med å realisere dette prosjektet. Først vil gruppen takke intern veileder ved HVL, Carsten Gunnar Helgesen, som har bidratt med tett og uvurderlig oppfølging til gjennomføringen av prosjektet. Videre vil gruppen takke ekstern veileder ved Adfectus AS, Christer Kilavik, som har vært en nøkkelperson for utformingen av modulen og alltid vært lett tilgjengelig når det trengs bistand. I tillegg ønsker gruppen å takke daglig leder, Randi-Marie Pedersen, og resten av Adfectus-teamet, for muligheten til å bidra i deres svært viktige arbeid. Til slutt vil gruppen takke alle som har vært med å teste produktet og andre som har støttet på ulikt vis.

ORDBOK

<i>Begrep</i>	<i>Beskrivelse</i>
<i>Gruppen</i>	Studentene som skriver oppgaven
<i>Teamet</i>	Gruppen i tillegg til ansatte hos Adfectus
<i>Modul</i>	Oppgaven skal utvikle en ny modul til Adfectus appen som i dag består av 6 moduler. Kan betraktes som et minste kravs produkt.
<i>MVP</i>	Minimum viable product. (Norsk: Minste kravs produkt)
<i>System</i>	En helhetlig betegnelse av hele Adfectus appen.
<i>Produkt</i>	En helhetlig betegnelse av modulen som skal utvikles.
<i>IDE</i>	Integrert utviklingsmiljø.
<i>Backend</i>	Serverside av systemet.
<i>Frontend</i>	Klientside av systemet (modulen).
<i>GIT</i>	Kodeversjoneringsystem.
<i>ER</i>	Entitetsrelasjon. (Entity Relation)
<i>NoSQL</i>	Not only SQL/Non-SQL. Databasemodell som ikke er relasjonsbasert.
<i>API</i>	Programmeringsgrensesnitt (Application Programming Interface).
<i>Backlog</i>	Liste med oppgaver tilknyttet utviklingen av modulen
<i>Scroll</i>	Side som kan navigeres med rulling.
<i>View(side)</i>	«Sidene» i appen som brukeren navigerer
<i>URL</i>	Uniform Resource Locator. Nettadresse
<i>State(tilstand)</i>	Verdien til et objekt eller variabel.
<i>Store(varehus)</i>	Lokalt datavarehus for håndtering av tilstander.
<i>Stabel</i>	Abstrakt datastruktur som følger sist inn, først ut prinsippet
<i>HTML</i>	Hyper Text Markup Language er et markeringsspråk for formattering av innhold i nettleser.
<i>JavaScript (JS)</i>	Programmeringsspråk.
<i>React-Native</i>	Rammeverk til JavaScript for utvikling av mobilapplikasjoner for iOS og Android.
<i>Discord</i>	Kommunikasjonsplattform.
<i>Notion</i>	Agilt utviklingsverktøy.
<i>Dropbox</i>	Skylagrings program.
<i>GitHub</i>	Skybasert hosting tjeneste for å håndtere Git depoter (repositories)

INNHOLDSFORTEGNELSE

1	INNLEDNING	1
1.1	Kontekst	1
1.2	Motivasjon	1
1.3	Prosjekteier	2
1.4	Problembeskrivelse og mål	3
1.5	Oppbygning av rapporten	4
2	PROSJEKTBEKRIVELSE	5
2.1	Praktisk bakgrunn	5
2.1.1	Tidligere arbeid	5
2.1.2	Initielle krav	5
2.1.3	Initiell løsningsidé	6
2.2	Systemarkitektur	7
2.3	Avgrensninger og rammer	8
2.4	Ressurser	9
2.4.1	Adfectus	9
2.4.2	Beitostølen Helseportsenter	9
2.4.3	Høgskulen på Vestlandet	9
2.4.4	Andre ressurser	10
2.5	Litteratur om problemstillingen	10
3	DESIGN AV PROSJEKTET	11
3.1	Arv av design	11
3.1.1	Drag and drop	13
3.2	Forslag til løsning	13
3.2.1	Løsningsforslag 1	14
3.2.2	Løsningsforslag 2	15
3.2.3	Løsningsforslag 3	16
3.2.4	Diskusjon av alternativene	17
3.3	Valgt løsning	17
3.4	Valg av verktøy	18
3.5	Prosjektmetodikk	18
3.5.1	Prosjektplan	18
3.5.2	Utviklingsmetodikk	18
3.5.3	Risikovurdering	19
3.6	Evalueringsplan	19
3.6.1	Verifikasjon	20

3.6.2	Validering.....	20
4	DETALJERT LØSNING.....	21
4.1	Inndeling av modulen	21
4.1.1	Navigasjon.....	22
4.1.2	Dagsoversikt.....	24
4.1.3	Måltidsbyggeren.....	25
4.2	Design og utforming av brukergrensesnitt	28
4.3	Lagring av data	28
4.4	Integrering med resten av appen.....	29
4.5	Tilrettelegging og Happtikk.....	32
5	RESULTATER	33
5.1	Evalueringsmetode	33
5.1.1	Enhetstest og modultest.....	33
5.1.2	Integrasjonstest og systemtest	33
5.1.3	Brukertest	33
5.2	Evalueringsresultat	34
5.3	Prosjektresultat	35
6	DISKUSJON.....	37
6.1	Prosjektgjennomføring	37
6.2	Påvirkninger på resultatet	37
6.3	Analyse av resultat.....	38
7	KONKLUSJON OG VIDERE ARBEID.....	39
7.1	Oppfylte delmål og krav	39
7.2	Delvis oppfylte delmål og krav	39
7.3	Hvorfor ikke alle mål ble oppfylt	39
7.3.1	Rapportskriving vs. koding	39
7.3.2	Ny teknologi.....	40
7.4	Videre arbeid	40
7.5	Lærdom andre kan ha av dette prosjektet.....	40
7.5.1	Opplæring i teknologi.....	40
7.5.2	Struktur på møter.....	40
7.5.3	Balanse mellom skriving og koding.....	41
8	REFERANSER.....	43
9	VEDLEGG.....	45

1 INNLEDNING

1.1 Kontekst

For mange pasienter og pårørende er kommunikasjonen med helsetjenesten krevende, særlig for barn. Informasjonen gis på en gammeldags måte som er generisk og lite tilpasset pasienten. Selv om vi lever i et samfunn med rask digital utvikling, henger den etter på denne fronten. For barn kan det være vanskelig å forstå informasjonen de får skriftlig, og samtaler med behandlere føles gjerne fremmed og skremmende. Selv om det går via pårørende, er det ikke alltid forståelig for barnet (Thorsen, 2019).

Psykiater Paul Joachim Bloch Thorsen er en spesialist i Barne- og Ungdomspsykiatri, og han jobber som Seksjonsleder på Barneposten ved Haukeland Universitetssykehus. Han har flere innovasjonsprosjekter innen bruk av personlig visuell kommunikasjon med barn (*Paul Joachim Bloch Thorsen, 2023*).

Randi-Marie Pedersen skrev en masteroppgave i 2021 basert på problemstillingen nevnt øverst i kapitlet. Da lagde hun en prototype for en app som skal gjøre det lettere for barn i helsetjenesten. Den utforsker bruken av personlig tilpasset visuell informasjon (Pedersen, 2021). Denne oppgaven ga opphav til bedriften Adfectus og appen deres med samme navn. Thorsen har hatt en sentral rolle med hans arbeid som oppgaven baseres på, og han var med å starte Adfectus som medeier. Her har de fått med seg pilotkunden Beitostølen Helseportsenter (BHSS, heretter omtalt som bare Beitostølen eller kunden), som bidrar til utviklingen av appen gjennom testing og tilbakemeldinger. Det er også et samarbeid med Røde kors Haugland Rehabiliteringssenteret og Haukeland Universitetssykehus, men deres rolle er i ikke like involvert.

Beitostølen er en rehabiliteringsinstitusjon innenfor spesialhelsetjenesten. De åpnet i 1970, og er et samarbeid mellom Erling Stordahl og sentrale myndigheter. (BHSS, 2023)

De gir behandlingstilbud til barn, unge og voksne med ulike diagnoser, funksjonsnedsettelse og funksjonsnivå. Og hvert år har de rundt 900 pasienter på opphold på senteret. Tilbudet dekkes av det offentlige, uten kostnad for brukerne. (BHSS, 2023)

Beitostølen benytter en sammensetning av tilpasset fysisk aktivitet med medisinsk, pedagogisk og sosialfaglig veiledning med oppfølging. Hvert opphold organiseres av en rekke fagpersoner, fra rideinstruktører og fysioterapeuter til leger. For alle pasienter er det viktig at de føler at de er aktivt med i utbygningen av behandlingsplanen, og at innholdet i behandlingen kan overføres til egen hverdag etter endt opphold. (BHSS, 2023)

Beitostølen har en avdeling med barn i alderen 6-12 år som er under opphold i normalt 3-4 uker. For å møte deres behov og bedre hjelpe dem, vil Beitostølen at barna og deres foresatte skal ta en aktiv rolle i egen behandling.

Adfectus-appen passer Beitostølenes behov for å oppnå dette. Ved oppstart av dette prosjektet har den 6 moduler. Hver av dem skal hjelpe med ulike scenarier pasientene møter før, under og i etterkant av innleggelse. Appen inneholder blant annet en pakke- og en aktivitetsplanlegger. Neste modul Adfectus og Beitostølen ønsker er en matplanlegger.

1.2 Motivasjon

Arbeidsmengden til sykepleiere og leger i det norske helsevesenet er langt over deres kapasitet, så de må prioritere hva de skal bruke tid og ressurser på. Ofte får ikke pasientene den oppfølgingen de

trenger for best mulig og raskest behandling. I en utredning (NOU 2023: 4) gjort for regjeringen mellom 2017 og 2023 kom det frem at det største problemet med helse- og omsorgstjenesten i Norge er ikke mangel på personell, men bruken av dem. Personellveksten har i lang tid vært betydningsfullt høyere enn befolkningsveksten, og er dermed ikke bærekraftig. Andelen blant alle sysselsatte i Norge er tredoblet siden tidlig 1970-tallet. Så derfor må prioriteringen for helsepolitikere være å dempe denne veksten gjennom å investere i relevante løsninger.

«Innovasjon, ny teknologi og digitale hjelpemidler i helse- og omsorgstjenestene har bidratt til bedre kvalitet og større omfang av tjenester.» (NOU 2023: 4, s. 19). Digitale verktøy kan bidra til å forenkle oppgaver for helsearbeidere og i noen tilfeller automatisere dem fullstendig, fra enkel kartlegging av nye pasienter til avansert kreftdiagnostikk ved hjelp av maskinlæring. Ved å gi pasientene tilgang til digitale plattformer kan de selv oppgi relevant informasjon, uten behov for assistanse fra helsepersonell. Dette kan frigjøre verdifull tid og ressurser for helsevesenet og bidra til en mer effektiv pasientbehandling.

Dagens barn og unge har et intuitivt forhold til digitale løsninger og den digitale arena, disse gir barn nye muligheter å uttrykke seg på. (Pedersen, 2021) Dette er spesielt nyttig for barn med kommunikasjonsvansker, fordi det gir dem en plattform de kan uttrykke seg på. Sammen med den økte digitaliseringen av helsevesenet følger det også frustrasjon på grunn av krav og økt stressfaktor, som er til ulempe for helsepersonell i form av tilstedeværelse og forståelsen ovenfor barn. (Pedersen, 2021).

For Beitostølen er sunne matvaner viktig å ta tak i og jobbe med hos pasienter; det gjelder for de fleste helsetjenester. Når et barn kommer til avdelingen, gjennomføres det en introduksjonssamtale med både lege og sykepleier for å kartlegge behov og mål for oppholdet. Allergier og intoleranser blir tatt opp ettersom det er nødvendig. Imidlertid blir ikke matpreferanser utover dette vurdert som en del av introduksjonssamtalen.

For mange barn er det en stor overgang å skulle bo i fremmede omgivelser. Å legge så mye som mulig til rette for dem vil gjøre det betydelig lettere for dem å slappe av og fokusere på behandlingen.

De fleste barn har sterke matpreferanser. Å gi dem eierskap over hva de skal spise kan gjøre mat mer lystbetont, noe som er spesielt viktig for dem med ulik problematikk rundt mat. Å konkretisere en måltidsplan vil gjøre det lettere å følge opp barnet.

1.3 Prosjekteier

Som nevnt tidligere, startet Adfectus som en masteroppgave fra Høgskulen på Vestlandet der Randi-Marie Pedersen tok en master i Innovasjon og entreprenørskap. I dag er hun daglig leder for bedriften med seks andre ansatte, og de er en del av Eitri Medisinsk Inkubator som holder til ved Haukeland sykehus i Bergen. På Adfectus sine nettsider står det:

Adfectus utvikler et banebrytende og etterlengtet kommunikasjonsverktøy for barn i helsevesenet, ved å la dem få tilgang til sin egen helsetilstand. Ved bruk av gamification og illustrative elementer styrker vi myndiggjøringen av barn og unge, uavhengig av fysiske og psykiske utfordringer. Teamet vårt har bred erfaring med hensyn til barns helse og behov, programvareutvikling og innovasjon. Sammen har vi skapt en unik tilnærming som prioriterer brukernes perspektiv. (Adfectus, 2022)

1.4 Problembeskrivelse og mål

Å kartlegge alle matpreferanser til hver pasient og lage en plan for dem kan være svært tidkrevende, og dette er ikke noe sykepleierne kan prioritere. Men om barnet kan gjøre det på egenhånd eller sammen med en foresatt, blir det en overkommelig oppgave.

Målet med dette prosjektet er en måltidsplanlegger-modul i Adfectus-appen som skal hjelpe barn med å etablere gode matvaner, med minimal ressursbruk fra helsepersonell.

Måltidsplanleggeren skal først og fremst være en kalender der barnet kan legge inn hvert måltid for ukene, og de skal ha mulighet til å velge det de ønsker å spise fra en liste av muligheter. Den skal fungere på en generell basis, slik at den ikke er avhengig av en rullerende meny på Beitostølen – eller tilsvarende institusjoner. Middagene på Beitostølen er ganske fastsatt med lite rom for endringer, så ønskene skal fortrinnsvis gjelde for resten av måltidene og eventuelle mellommåltider. Disse består i stor grad av faste produkter som blant annet skiver, pålegg og frukt.

Det er viktig at modulen skal kunne brukes i etterkant av innleggelsen for å opprettholde gode vaner. Her vil det være naturlig at middager skal kunne velges på samme måte som resten av måltidene. Det er også ønskelig at listen av tilgjengelige produktet skal kunne utvides av brukerne. Dermed blir det en liten separasjon av funksjonalitet under opphold og etter.

Om det blir nok tid skal Adfectus sin administratortnettside utvides til å inkludere matplanleggeren. Den skal inneholde informasjon som behandlerne kan bruke for å få oversikt over pasientene, men det er ikke tenkt at den skal overvåke dem på et individuelt nivå.

Delmål som prosjektet har:

- Det må lages en måltidsbygger. Her kan brukerne sette sammen ett og ett måltid fra en liste over mat- og drikkevarer. Dette vil være én av de to mest sentrale komponentene i modulen.
- En oversikt for kommende måltider er den andre sentrale komponenten. Fra den skal brukeren også kunne legge til nye og redigere eksisterende måltider. Først skal det gjøres for de enkelte måltider, og så skal de settes sammen for flere dager og uker. Det ender opp med ukeplaner, med syv dager per uke, og fire måltider per dag.
- Når funksjonaliteten til de to overnevnte punktene er på plass, må de tilpasses designmessig slik at de samsvarer med resten av Adfectus-appen.
- Deretter må tilstandshåndtering implementeres for å sikre at planene blir lagret lokalt på mobilen eller nettbrettet når appen lukkes.
- Det skal lages en brukeropplæring for modulen. Når dette målet er oppnådd, skal modulen virke i enkel grad.
- Utover dette er det ønskelig å få med flere funksjoner som gjenbruk av tidligere planer, poengsystem basert på hvor sunne måltid som lages er og en opplæringsmodul om ernæring.

1.5 Oppbygning av rapporten

Rapporten er delt inn i følgende kapitler:

Tabell 1: Inndeling av rapporten

Kapittel	Beskrivelse
1 INNLEDNING	Innledningen til prosjektet med beskrivelse av tematikk, mål og eier
2 PROSJEKTBEKRIVELSE	Beskriver i større grad bakgrunnen for prosjektet og hvilke krav den har. Den har også med avgrensninger, ressurser og hvilke litteraturer problemstillingen er bygget på
3 DESIGN AV PROSJEKTET	Presenterer tekniske detaljer til prosjektet, med ulike løsninger som er vurdert. Videre inneholder den hvordan teamet jobbet med utviklingen og hvordan resultatet skal evalueres
4 DETALJERT LØSNING	Uttrykker i dybden hva som er laget, og hvordan de ulike delene er utviklet
5 RESULTATER	Beskriver i detalj hvordan resultatene er evaluert, og hva som er endelig resultat
6 DISKUSJON	Drøfter sluttproduktet sett fra gruppens og produkteiers perspektiv, men også i en større, utenforstående sammenheng.
7 KONKLUSJON OG VIDERE ARBEID	Konklusjonen av hvorvidt problemstillingen har blitt løst, og hva som kan gjøres videre på prosjektet
8 REFERANSER	Liste med referanser som er brukt i rapporten
9 VEDLEGG	Alle støttedokumentene rapporten er bygget på. Inneholder Retningslinjer for kode og versjonskontroll, Prosjekthåndbok, Visjonsdokument og Kravdokument

2 PROSJEKTBESKRIVELSE

2.1 Praktisk bakgrunn

Dette delkapittelet omhandler bakgrunnen for prosjektet og tidligere arbeid som prosjektet bygger videre på.

2.1.1 Tidligere arbeid

I forkant av denne oppgaven ligger det to tidligere masteroppgaver.

- Designing an application for creating personalized comics aimed for the Children and Youth Clinic at Haukeland University Hospital (Syslak, 2019)
- Samproduksjon i psykisk helsevern for barn og unge: En studie av pasientsentrering gjennom digitale helseinnovasjoner. (Pedersen, 2021)

Den førstnevnte er mindre relevant fordi den omhandler bare personalisering av tegneserier for bruk som informasjonsformidling om behandlingsløp og siden dette prosjektet ikke skal stå for design elementer (design elementer: bilder og andre spesiellagde grafiske elementer brukt i appen), men funksjonalitet. Den er fortsatt noe relevant innen design og utforming ettersom den bruker relevante prinsipper. Den markerer starten på Adfectus. Uten oppgaven hadde ikke Pedersen utført sin studie, og dermed aldri grunnlagt Adfectus. Det er også her samarbeidet med Paul Joachim Bloch Thorsen, spesialist i Barne- og Ungdomspsykiatri, startet. Han har vært en nøkkelperson i utarbeidningen av Adfectus med sin kompetanse som psykiater for barn.

Adfectus AS er videreutviklet fra den sistnevnte masteroppgaven, skrevet av gründer Randi-Marie:

Digitalisering av offentlige tjenester blir generelt ansett som en betydelig fasilitator for involvering av kunder i leveranseprosesser. Gjennom bruk av teknologiske løsninger og en tilnærming til samproduksjon beveger tjenester seg over mot individualiserte tilbud og prosesser. For at helsevesenet skal sikre en pasientsentrert helsemodell blir det derfor nødvendig å prioritere digitale helseløsninger, samt inkludering av pasienter gjennom en helhetlig og sømløs leveranse. (Pedersen, 2021. Side 1)

Denne masteroppgaven går igjennom punkter for mål og visjon som er svært viktige for dette prosjektet, som: Samproduksjon i offentlige tjenester, digitalisering og spillifisering av tjenester i psykisk helsevern, og pasientinvolvering. Som nevnt i 1.2 har dagens barn og unge et intuitivt forhold til digitale løsninger. Dermed er det nødvendig med en slik plattform utviklet for barn, samtidig som at den skal gjøre arbeidet til helsepersonell mindre krevende. Dette er grunnlaget for masteroppgaven til Randi-Marie og Adfectus. Dette skal prosjektet som beskrives i denne rapporten, fortsette på, med en utvidelse av appen deres. En betydelig del av barn som er i helsetjenesten har kommunikasjonsvansker, og dermed må Adfectus appen tilpasses dem.

2.1.2 Initielle krav

De initielle kravene er utformet sammen med oppdragsgiver og BHSS. Oppdragsgiver krevde også at modulen skal kodes i React-Native ettersom resten av applikasjonen den skal integreres i også er skrevet i React-Native. Applikasjonen er bare utformet for landskapsvisning og dette skal modulen vår også.

Tabell 2: Krav for et minsteprodukt av modulen. Se vedlegg 4 for mer utfyllende tabell

Krav	Beskriving	Prioritering
Planlegge uka	Planlegge måltid for den kommende uka. Brukeren skal kunne planlegge måltidene for uka (frokost, lunsj, middag, kvelds og evt. mellommåltider), fra mandag til søndag.	1
Inndeling av plan	Planleggeren skal inndeles i ukenummer. Brukeren skal kunne endre ukenummer for den aktuelle planen.	2
Lagring av plan	Den aktuelle uka sin plan skal lagres lokalt på enheten til brukeren ved å trykke på en knapp i appen.	1
Plan-logg	Lagre gamle/utdaterte planer. Tidligere planer skal lagres lokalt på enheten til brukeren.	2
Import av tidligere planer	Tidligere planer skal kunne importeres inn i den aktuelle ukas plan.	2
Flere planer	Brukeren skal kunne lage flere planer og kunne lagre dem lokalt. Brukeren skal kunne endre, slette og kopiere planar.	4
Variasjonslogikk	Brukeren kan ikke legge inn det samme måltidet på alle måltid for en dag. Yoghurt kan ikke være på både frokost, lunsj, middag og kvelds.	5
Ernæringsopplæring	En side hvor brukere får enkel informasjon om ernæring og omliggende. Eksempel: Sjokolade er usunt; Frukt er sunt.	2
Sletting	Brukeren må bekrefte om en plan skal slettes.	2
Varsler	Varsle bruker om kommende måltid.	3
Avatar	Brukerens avatar skal brukes i modulen der det er naturlig.	2
Tilrettelegging	Brukeren burde få feedback ved lyd og haptikk.	1
Brukeropplæring	Det skal lages en «How-to» skjerm for modulen, som viser hvordan modulen skal brukes, og generell informasjon rundt modulen. Denne er ment som en bruksanvisning.	1
Utforming	Målgruppen til appen er begrenset og må derfor designes med denne i hovedfokus. Utformingen skal også være i henhold til Affectus sine retningslinjer for design.	1

2.1.3 Initiell løsningsidé

De viktigste kravene er: Planlegge uka, lagring av plan, tilrettelegging og brukeropplæring. Modulen har en startside hvor man kan navigere til forskjellige planer. På hver plan kan man videre navigere til hver enkelt dag hvor man kan endre på planen. Dersom man har gjort endringer på en plan skal den lagres automatisk. Modulen skal ha tilrettelegging for personer med nedsatt funksjonsevne, dette inkluderer tilbakemelding ved haptikk og lyd slik at det skal være lettere å bruke appen. Det skal lages en side hvor brukere får opplæring i hvordan de skal bruke modulen, dette inkluderer alt som brukeren trenger for å få full utnyttelse av modulen.

Målgruppen til modulen er barn i alder 6-12 år. Dermed er det viktig med et grensesnitt som de intuitivt kan samhandle med. Modulen skal være minimalistisk i form av mindre knapper og funksjonalitet som er innlemmet i andre handlinger. F.eks. når man har endret en plan blir den automatisk lagret slik at brukeren ikke trenger å gjøre det. Det ble utarbeidet tre ider basert på de initielle kravene. Noen elementer er med i alle løsningsideer deriblant lagring og utviklingsstruktur.

2.1.3.1 Tabell-løsning

En tabell-løsning der hele planen er en ukentligkalender hvor bruker kan se hele planen for en gitt uke på én side, og når de trykker på en dag kommer de videre til en side hvor det kan planlegges måltid for denne dagen. Denne er kompakt både i utforming og utvikling.

2.1.3.2 Kort-basert

Løsningen består av tre deler. En forside hvor man kan raskt se relevant informasjon, en tabell side hvor informasjon om gitt dag, og en planleggingsside hvor man kan lage et måltid for en gitt dag.

Forsiden er lett og viser bare informasjon som bruker trenger raskt tilgang til. Fra forsiden skal de kunne navigere seg inn til en side hvor de kan se alle måltidene for dagen.

Tabell siden skal vise en en-dimensjonal tabell hvor dagen er den eneste raden og kolonnene er måltidene. Her skal man kunne navigere seg videre til planleggingssiden. Hver tabell er vist som et kort hvor man kan bytte kort (bytte uke).

Planleggingssiden er den siste siden hvor man skal kunne planlegge måltid og måltidstype for en gitt dag.

2.1.3.3 Lett kort-basert løsning fordelt på to sider.

Løsningen kan sees på som en kombinasjon av de to øvrige. Modulen er delt i to, en forside hvor man kan se alle måltid for en gitt dag, og en planleggingsside hvor man kan planlegge et måltid for en gitt dag.

På forsiden skal man se all informasjon om en gitt dag. Man skal kunne velge hvilken plan og hvilken dag man vil se informasjon om. Her ifra skal man kunne navigere seg til planleggingssiden. På planleggingssiden skal man kunne planlegge måltidet for den gitte dagen.

2.2 Systemarkitektur

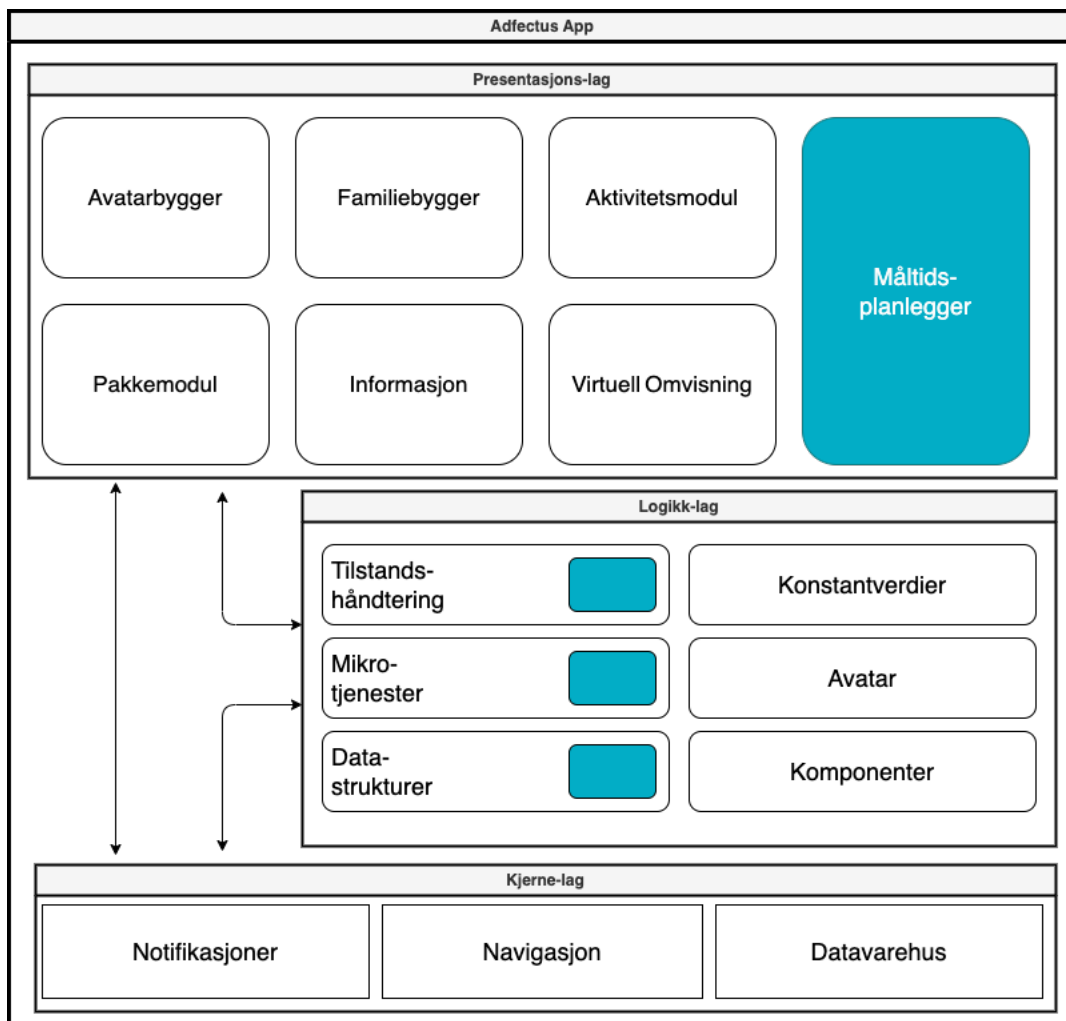
Den nåværende arkitekturen består av tre lag (se figur 1).

Kjernen inneholder nødvendig funksjonalitet for appen. Logikk inneholder all gjenbrukbar kode. Presentasjon inneholder all kode for de individuelle komponentene.

I kjernen på prosjektet ligger skallet til appen. Her ligger notifikasjonskomponenten som skal brukes i prosjektet. Rutermodulen setter opp navigering i appen, og modulen skal integreres her. Store er et datavarehus brukt for å håndtere tilstander i appen.

I Logikklaget ligger all kode som skal gjenbrukes. I tilstandshåndtering ligger det funksjoner som jobber opp mot Store.

Presentasjonslaget inneholder alle sider som skal vises på skjermen til bruker. Måltidsplanleggeren inneholder mesteparten av prosjektet.



Figur 1: Adfectus nåværende systemarkitektur. Felt markert i blått er en del av prosjektet.

2.3 Avgrensninger og rammer

Dette Prosjektet bygger en ny modul for Adfectus appen, og dermed ligger det flere avgrensninger med tanke på teknologi, kodestandard og utforming. Et mål med prosjektet er å bygge modulen dynamisk slik at Adfectus-teamet kan bygge videre på den etter at prosjektet er fullført. Prosjektet har klare retningslinjer for kodestandard og arbeidsmetodikk utarbeidet av Christer Kilavik og Randi-Marie Pedersen fra Adfectus-teamet (se vedlegg 2). Ettersom målet til prosjektet er å innlemme modulen inn i Adfectus appen, er det flere retningslinjer innen utforming som prosjektet burde følge for å få en sømløs integrasjon både for kode og design.

Det er begrenset mulighet for testing i dette prosjektet, delvis på grunn av pilotkunden og målgruppen. Beitostølen Helse- og sportsenter er valgt som pilotkunde og har behandlingslokaler på Beitostølen. Dette gjør det vanskelig å gjennomføre omfattende testing av produktet. Oppdragsgiver ønsket også å vente med brukertesting til produktet var nærmest ferdig utviklet. Dette skyldes erfaringer med målgruppen som viser at den kan ha vanskeligheter med å forstå at produktet ikke er ferdig når det testes på et tidlig stadium. Det er dermed begrensede muligheter for å få tilbakemeldinger fra brukerne i tidlig fase av utviklingen, men det vil være viktig å gjennomføre grundig testing før produktet lanseres for å sikre at det møter målgruppens behov og forventninger på best mulig måte.

Modulen som er utviklet i dette prosjektet har som mål å lagre relevant data lokalt på enheten til brukerne. Dette er en alternativ tilnærming til tradisjonelle systemer som lagrer brukerinput i databaser. Slike databaser krever normalt sett større sikkerhet mot dataangrep som injeksjon og lignende. I dette prosjektet anses ikke dette som relevant, da dataene som skal lagres, er begrenset til enheten den brukes på, og de inneholder kun informasjon om brukerens matvalg og er ikke personlig identifiserbar informasjon. Det er imidlertid viktig å bemerke at datasikkerhet generelt sett er en viktig faktor å ta hensyn til i utviklingen av digitale løsninger og spesielt i helsesektoren hvor sensitiv informasjon håndteres.

2.4 Ressurser

For å kunne best løse prosjektet, har gruppen fått tilgang til mange ressurser fra både Adfectus og Høgskulen på Vestlandet (HVL). Dette inkluderer blant annet arbeidsplass, møterom, teknologi og kommunikasjonsløsninger. I dette kapittelet presenterer vi de viktigste ressursene som inngår i prosjektet. Adfectus er gruppens største ressurs ettersom de er prosjekteier. Det legges også spesielt vekt på Beitostølen Helseportsenter ettersom de er pilotkunde til prosjektet. Teknologiske ressurser som prosjektet benytter blir beskrevet i kapittel 3.

2.4.1 Adfectus

Adfectus er oppdragsgiveren til prosjektet samt dets viktigste ressurs, og de bidrar med flere essensielle ressurser for prosjektet. De har bl.a. gitt gruppen tilgang til kontor plass og møterom ved Eitri Helseinkubator på Haukeland. Denne ressursen har gitt gruppen en fast plass hvor den kan sitte sammen og jobbe, samt en uvurderlig mulighet til holde direkte kommunikasjon sammen med teamet. Adfectus har også lånt studentene skjermer og iPad ved behov.

Den viktigste ressursen fra Adfectus er deres veiledning og kompetanse igjennom prosjektet. Leder for virtuell omvisning, Christer Kilavik, daglig leder Randi-Marie Pedersen og leder for mobilapplikasjon, Sondre L. Solberg, er viktige kunnskapsressurser for prosjektet. De står til rådighet med kompetanse, tilbakemeldinger og hjelp dersom det trengs.

Paul Joachim Thorsen er medgründer og klinisk ansvarlig i Adfectus, og stiller seg selv til rådighet med kompetanse, veiledning og testing der det er nødvendig. Testingen vil foregå på barneposten ved Haukeland sykehus.

2.4.2 Beitostølen Helseportsenter

Beitostølen Helseportsenter (BHSS) er pilotkunden til prosjektet og dermed en veldig viktig ressurs for prosjektet. De står til rådighet med kompetanse fra helsesektoren og testing av produktet. Kontaktpersonen fra BHSS er Berit Gjessing, og hun stilles til rådighet med kompetanse, tilbakemeldinger og testing av produkt der det trengs.

2.4.3 Høgskulen på Vestlandet

Høgskulen på Vestlandet (HVL) stiller med teknologiske og menneskelige ressurser til prosjektet. HVL tilbyr flere tjenester som prosjektet kan nytte. Blant annet grupperom, møterom, servere, programvare, kompetanse og veiledning. Det er verdt å nevne at Carsten Helgesen, intern veileder fra

HVL, er også en del av Adfectus-teamet og er en særs viktig ressurs for prosjektet, da spesielt innen akademisk skriving og prosjektplanlegging. Han gir veiledning kontinuerlig igjennom hele prosjektet.

2.4.4 Andre ressurser

Studentene stiller selv med datamaskiner, programvare og annet utstyr til prosjektet.

2.5 Litteratur om problemstillingen

Som en del av undersøkelsen for innledningen fikk gruppen tilsendt forelesningsnotater fra medgründer Paul Joachim Bloch Thorsen, *Safe and Better Communications with Personalized Visual Information - Making information understandable, motivating and engaging* (Thorsen, 2019). Den var nyttig for å gi en klar forståelse for hva som er utfordringene rundt kommunikasjon med barn i helsevesenet, og hvordan behandlere, ved hjelp av digitale verktøy, kan gjøre det lettere.

I forbindelse med opplæring fikk gruppen et dokument fra Adfectus med retningslinjer for koding, og rutiner ved versjonskontroll (vedlegg 2). Dette dokumentet vil bli brukt som et oppslagsverk gjennom prosjektperioden.

I dialog med veilederen ble det tilsendt en utredning av personellet i helse- og omsorgstjenestene, *Tid for handling: Personellet i en bærekraftig helse- og omsorgstjeneste*. Den ble brukt til å få et dybdebilde av hvilke utfordringer de har og kommer til å ha i årene som kommer, og hva som kan gjøres for å motvirke trenden med at de blir gradvis mer overarbeidet. Denne brukes for kontekst i kapittel 1.2.

3 DESIGN AV PROSJEKTET

Inspirasjonen for designet av prosjektet hentes fra moduler som allerede er innebygd i Adfectus-appen. Det er et sentralt poeng å speile brukerinteraksjon fra allerede eksisterende deler av appen, slik at måltidsplanleggeren blir godt inn i den totale brukeropplevelsen. Adfectus har satt krav til at måltidsplanleggeren følger gjeldende stil og uttrykk, men at det utenfor dette er frihet til å komme med nye designideer.

3.1 Arv av design

Tre deler av appen inneholder funksjonalitet og brukerinteraksjon som er aktuelt å arve for den endelige løsningen: Dørskilt-lageren, pakkemodulen og avatarbyggeren.

Dørskiltlageren

Dørskiltlageren tilhører en annen modul, og skal hjelpe brukeren å lage et dørskilt for oppholdet (figur 2). På høyre side er en vertikal scrollbar liste av bakgrunner for skiltet, og navnet til brukeren står på et skilt foran valgt bakgrunn. Brukergrensensnittet er svært grunnleggende, og har ingen overflødige knapper. Trykker man på «Tilbake», lagres dørskiltet og går tilbake til startmenyen for modulen.



Figur 2: Skjerm bilde fra dørskiltlageren. Pasientnavn vises på skiltet, og på høyre side er det en liste over bakgrunner som brukeren kan velge mellom.

Pakkemodulen

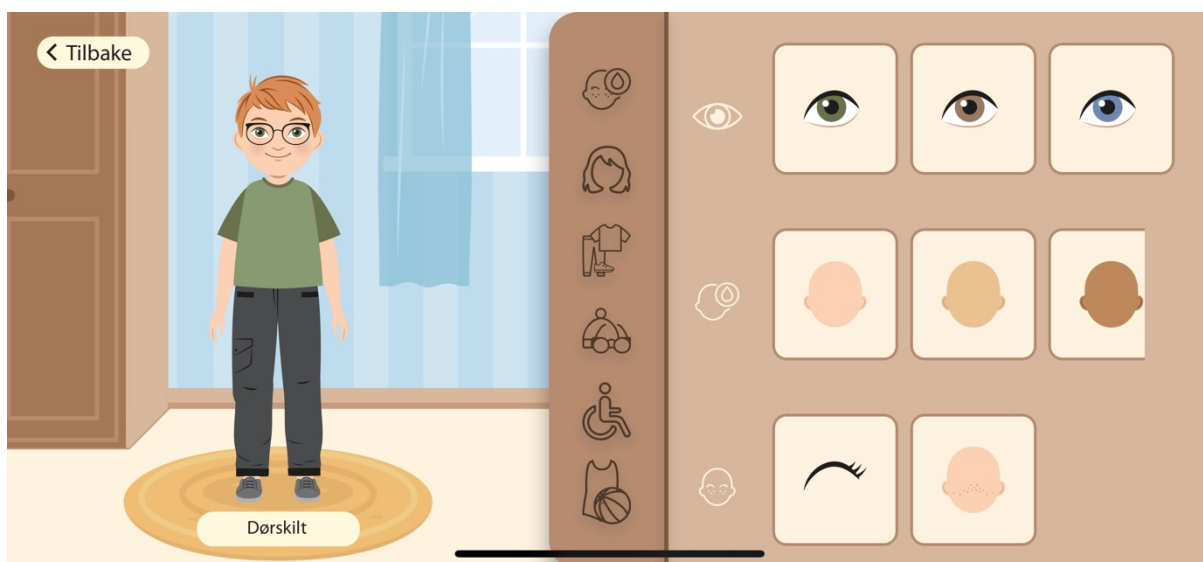
Pakkemodulen er laget for at brukere kan pakke virtuelt for oppholdet på behandlingsstedet. Her får brukeren mulighet til å dra inn diverse objekter i en koffert (figur 3), og på den måten bli påminnet om hva som kan være greit å ha med seg. Her brukes «drag and drop», som er en teknikk oppdragsgiver ønsker også skal brukes i måltidsplanleggeren (se kapittel 3.1.1).



Figur 3: Skjerm bilde fra pakkemodulen. Brukeren kan dra og slippe alle objekter som vises inn i kofferten, men blir vennlig stoppet når man prøver å pakke noe som ikke hører med på oppholdet, f.eks. kaffelageren på hylla.

Avatarbyggeren

Avatarbyggeren inneholder en meny på høyre side av skjermen, som lar brukeren konfigurere en avatar som blir med videre i de andre modulene (figur 4). Måten denne menyen er lagt opp på, vil være aktuelt å følge i utforming av menyer i måltidsplanleggeren.



Figur 4: Skjerm bilde av avatarbyggeren. Brukeren velger kategori til venstre i menyen, og får opp et utvalg konfigurasjonsmuligheter til høyre i menyen.

Design som er aktuelle for måltidsplanleggeren å arve er

- Tilbakeknappen, som fører brukeren tilbake til forrige view, samt lagrer valg brukeren har gjort. Dette inkluderer knappens plassering i bildet.
- Listemeny på høyre side, og oversikt på venstre side.
- Fargevalg.
- Brukerinteraksjon i form av drag and drop.

Fordeler ved å arve disse designvalgene, er at

- modulen reagerer likt på brukerinteraksjon sammenliknet med andre moduler i appen
- designet utelukker overflødig informasjon, og øker brukervennligheten for målgruppen
- fargene er gjenkjennbare, og bidrar til en sømløs integrasjon med resten av appen

Ulemper er at det overnevnte, til tross for at det er gode designvalg, også setter begrensninger for alternative designvalg i modulen. Det setter også spesifikke krav til prosjektgruppen om å sette seg inn i ukjente teknologier, noe som kan være svært tidkrevende når det må læres fra bunn.

3.1.1 Drag and drop

Drag and drop er en type brukerinteraksjon hvor elementer i brukergrensesnittet kan flyttes og slippes til ønsket eller forhåndsdefinert plassering ved å dra elementet over skjermen med fingeren (Techopedia, 2023). Fordelene med å benytte denne teknikken i måltidsplanleggeren er at

- barn i målgruppen får til å bruke den (Aziz, 2013)
- teknikken allerede er i bruk i andre moduler av Adfectus-appen
- gode leseferdigheter er ikke nødvendig for å bruke modulen

Ulemper er at

- det kreves en viss finmotorikk fra brukerne
- løsningen utløser et større behov for grafikk i modulen

Sistnevnte ulempe utløser kostnader ved bruk av ekstern designer.

Det er blitt vurdert at fordelene trumfer ulempene. Drag and drop er allerede omfattende implementert i andre moduler i Adfectus-appen, og ved å bruke denne teknikken i måltidsplanleggeren, bidrar dette til å bedre totalopplevelsen av appen.

3.2 Forslag til løsning

Løsningen på problemstillingen er på forhånd bestemt skal være en modul i Adfectus-appen. Det ligger klare føringer og ønsker for modulen. Likevel er det stor frihet for hvordan modulens funksjoner og egenskaper kommer til uttrykk, og det er dette som har vært tematikken i arbeidet med løsninger.

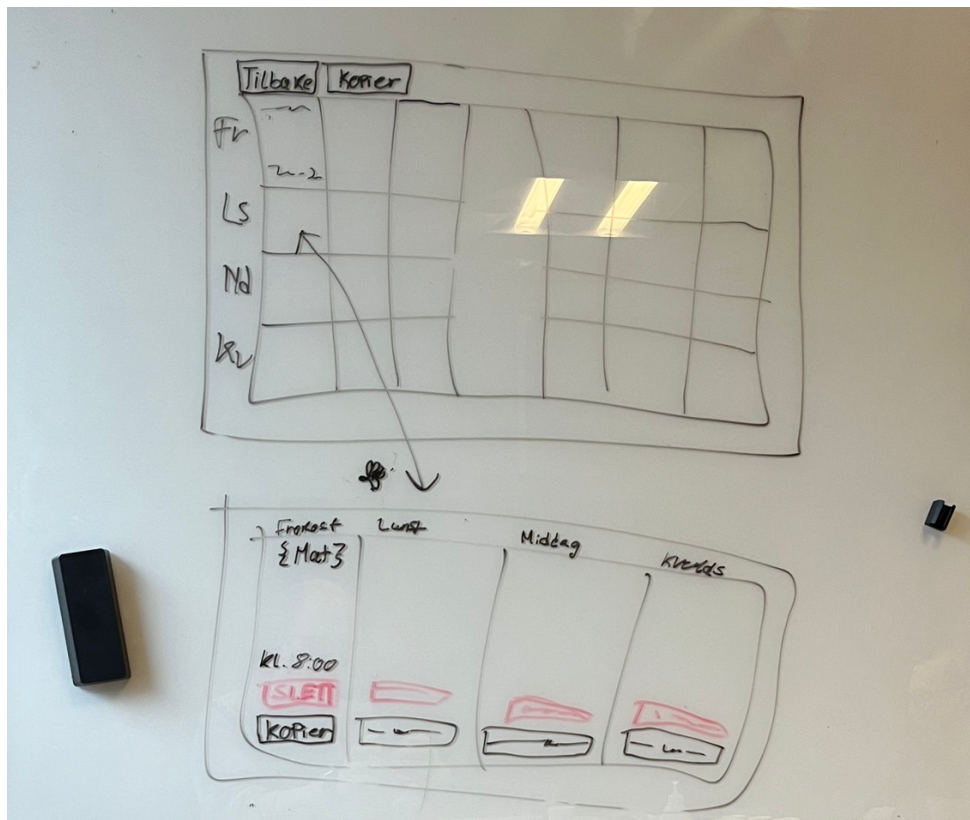
For best mulig å treffe prosjekteiers ønske for modulen, har det vært viktig å diskutere forskjellige struktureringer av modulen, la tidlige ideer få utvikle seg, og snakke om konsekvenser av løsningsvalg. Sammen med prosjekteier har det blitt gjennomført møter med idémyldring og skissering av skjermer og funksjonalitet.

To deler av modulen blir sentrale: En måltidsbygger og en måltidskalender. Måltidsbyggeren blir den delen av modulen hvor brukeren planlegger hvert enkelt måltid. Her blir drag and drop en sentral teknikk for å skape brukervennlighet og intuitiv design. Måltidskalenderen blir en oversikt over alle måltid som skal eller er blitt planlagt. Her vil brukeren kunne velge dager som skal vises, og gå inn på hvert enkelte måltid for å planlegge det.

Idéarbeidet har resultert i tre mulige løsninger for modulen, som alle oppfyller de krav som er spesifisert fra prosjekter.

3.2.1 Løsningsforslag 1

Løsningen baserer seg på at man får frem en ukeoversikt (figur 3.1) når man åpner modulen, med frokost (Fr), lunsj (Ls), middag (Md) og kvelds (Kv) satt inn på de klokkeslett som er bestemt av behandlingsstedet.



Figur 5: Løsningsforslag 1 (tegning fra idémyldring). Øverst i bildet vises ukeoversikten, hvor hver kolonne leder til dagsoversikten nederst i bildet.

Om brukeren skal planlegge måltid, trykker de på kolonnen som representerer dagen som skal planlegges, og brukeren blir tatt til en oversikt over dagen de har trykket på. Dagsoversikten inneholder hvert måltid, med valg for å kopiere eller slette måltid. På dagsoversikten vil brukeren kunne legge til matvarer ved bruk av drag and drop, og representasjoner av dette vil også dukke opp i ukeoversikten. Trykker man på «Tilbake» vil man bli tatt til en liste med ukeoversikter man kan gå inn på for å redigere. Det ble laget en enkel Figma-skisse (figur 6), for å få et inntrykk av hvordan plass på skjermen utnyttes i ukeoversikten.

iPhone 13 / 13 Pro - 1

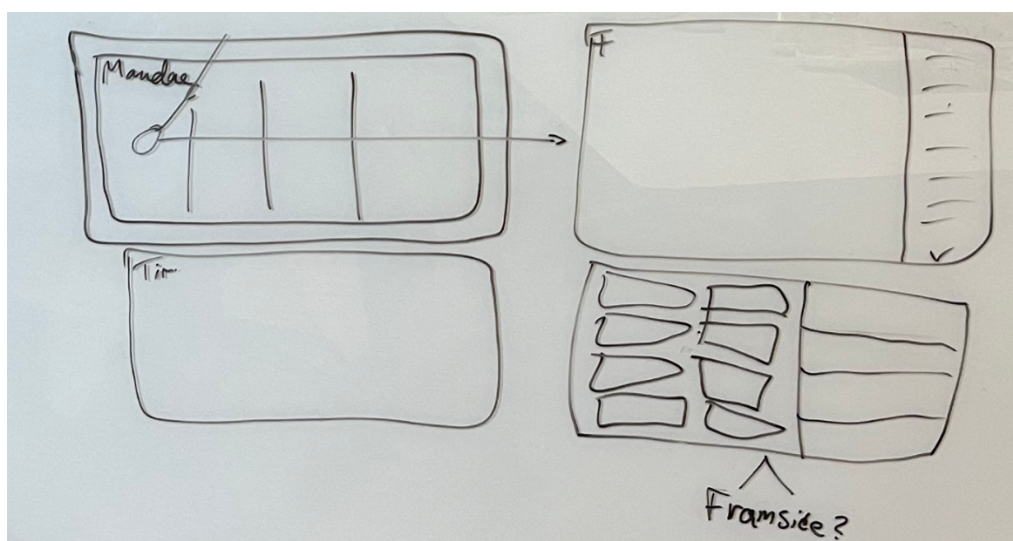
	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
Frokost kl 8.15							
Lunsj kl 12.15							
Middag kl 16.15							
Kvelds kl 19.15							

Figur 6: Ukeoversikt (skjerm bilde fra Figma). Hver rute inneholder valgte matobjekter for hvert enkelt måltid.

Et av kravene fra Adfectus er at brukeren av appen skal kunne hente frem tidligere måltidsplaner og bruke dem på nytt. Med løsningsforslag 1 vil brukeren også kunne kopiere dagsplaner for måltid og sette dem inn på ønsket dag.

3.2.2 Løsningsforslag 2

Løsningsforslag 2 sentrerer rundt virtuelle kort med dagsoversikt over måltidene (figur 7). Brukeren scroller vertikalt for å se og redigere måltidsplaner for andre dager. Denne løsningen tar i bruk skjermen på en annen måte enn i løsningsforslag 1, ettersom startskjermen her viser en dagsoversikt, og ikke en ukeoversikt.



Figur 7: Løsningsforslag 2 (tegning fra idémyldring). Øverst til venstre er forslag til dagsoversikt for mandag, som går til et enkelt måltid (øverst til høyre). Nederst til høyre er forslag til en startskjerm eller forside som møter brukeren når modulen åpnes.

En alternativ startskjerm for denne løsningen er en skjerm som inneholder knapper for forskjellige valg i tilknytning til planlegging av måltider, for eksempel «Vis ukeplan», «Vis dagsplan» og «Middagsmeny».

På dagsoversikten vil brukeren kunne trykke på måltidet de vil redigere, og dermed bli tatt videre til en skjerm hvor de drar matobjekter inn fra en sidemeny, ved bruk av drag and drop.

3.2.3 Løsningsforslag 3

Løsningsforslag 3 er en utvikling av løsningsforslag 2.



Figur 8: Løsningsforslag 3 (skjerm bilde fra Figma). Alle måltid for valgt dag vises som en liste av kort. Ønsker man å se måltider for en annen dag, velger man fra listen til høyre.

Også dette designet sentrerer rundt en startskjerm som viser måltid for dagen i dag, bare at man her har måltidskort som viser hva hvert måltid inneholder, og menyen på høyre side viser dagene i gjeldende uke. Trykker man på en annen dag i menyen til høyre, vil måltidene til valgt dag vises.

På samme måte som i løsningsforslag 2, vil man komme inn i Måltidsbyggeren dersom man trykker på et av måltidene.



Figur 9: Måltidsbyggeren (skjerm bilde fra Figma). Brukeren drar inn matobjekter fra menyen på høyre side, og slipper det i oversikten til venstre.

3.2.4 Diskusjon av alternativene

To premisser har vært sentrale i vurderingen av løsningsforslagene: Aldersgruppe på brukerne, og skjermstørrelse. 1) Aldersgruppe på brukerne er fra Adfectus' side definert til å være barn fra 6 til 12 år. 2) Enhetene som appen brukes på vil variere mellom mobiltelefoner og nettbrett, og det vil være viktig å skape en løsning som fungerer godt på varierende skjermstørrelser.

Løsning 1 har en tabell-liknende ukeoversikt som startskjerm, som inneholder 7 kolonner med 4 rader. Dette resulterer i 28 ruter med informasjon om måltid, noe som kan oppleves som en overveldende mengde informasjon, uavhengig av skjermstørrelse. På mobilskjermen blir grafikken spesielt liten. En slik startskjerm etterlater heller ikke tilstrekkelig plass til andre elementer på skjermen. Likevel kan ukeoversikten bli en nyttig del av modulen, dersom brukeren vil se måltidene for uka i sin helhet. Det kan også være nyttig med en funksjon som lar dem eksportere denne oversikten og sende til foreldre eller behandlere som PDF- eller bildefil.

Løsning 2 og 3 viser dagsplan for gjeldende dag som startskjerm. Dette bidrar til å gjøre informasjonsmengden som møter brukeren mer fordøyelig. For disse løsningsforslagene kan brukeren oppsøke mer informasjon, i stedet for at all informasjon presenteres på startskjermen. Løsningene gir mer luft i designet, og det blir bedre plass til knapper og andre nødvendige elementer.

3.3 Valgt løsning

Ut fra et helhetlig brukerperspektiv, har løsningsforslag 3 blitt valgt som løsning for modulen. Dette utelukker ikke at elementer fra løsningsforslag 1 og 2 inkluderes i utviklingen.

Grunnlaget for beslutningen ligger i at løsningsforslag 3 bedre møter de to hovedpremissene for bruk av modulen. Skjermutnyttelsen er mer effektiv, romslig og har en mindre overveldende mengde informasjon på hvert skjermbilde. Ukeoversikten har blitt lagt på is inntil videre. Etter diskusjon i prosjektgruppa er det usikkerhet om den i det hele tatt finnes brukstilfeller for barn i alderen 6 til 12

år. Det er også usikkerhet om barn i denne alderen klarer å forholde seg for planer flere uker frem i tid. Dette utelukker likevel ikke at en ukeoversikt blir lagt til som en fremtidig funksjon i modulen.

Utfordringer med valgt løsning

Løsningsforslag 3 deler modulen mer opp, noe som kan gjøre den mer uoversiktlig og komplisert å bruke i starten. Det vil bli prioritert å teste modulen på barn i målgruppen, og evaluere brukernes opplevelser. Testingen burde finne sted i tidlige faser av utviklingen, for å avdekke blant annet irrelevant funksjonalitet og lite treffende brukergrensesnitt.

Det vil, etter all sannsynlighet, bli anledning for deler av teamet å besøke pilotkunden, og i den sammenheng gjennomføre nødvendige tester.

3.4 Valg av verktøy

Adfectus-appen utvikles i rammeverket React Native, i kombinasjon med det åpne rammeverket Expo. Det blir derfor naturlig for teamet å utvikle måltidsplanleggeren ved hjelp av denne teknologien. Ved å bruke React Native, utelukkes utvikling i de native språkene for iOS og Android, men prosessen forenkles betraktelig ved å benytte teknologien som allerede brukes i bedriften.

Visual Studio Code brukes som tekstredigeringsverktøy med utvidelser som gjør den nærmest til en IDE, GitHub brukes til versjonskontroll, og Notion brukes i administrasjon av prosjektet internt i teamet.

Kommunikasjon med Adfectus foregår muntlig på kontoret, og digitalt på en Discord-server eller per e-post. Kommunikasjon med pilotkunden foregår per e-post, og ved primært digitale møter.

Prosjektet følger et veiledende Gantt-diagram (se kapittel 3.5.1). Diagrammet bidrar til å holde den overordnede oversikten i prosessen, og skildrer iterasjonene i grove trekk.

3.5 Prosjektmetodikk

Med en gruppe bestående av tre personer, blir det avgjørende å holde struktur på arbeidsoppgavene for å unngå dobbeltarbeid og misforståelser på oppgavefordeling.

3.5.1 Prosjektplan

En prosjekthåndbok er blitt opprettet for å samle informasjon om prosjektets fremgang, og for å holde rede på innholdet i iterasjonene. Et Gantt-diagram brukes som prosjektplan, og inneholder tidsfestede milepæler, iterasjoner og tester. Både prosjekthåndboken og Gantt-diagrammet blir oppdatert fortløpende, slik at kartet stemmer overens med terrenget.

3.5.2 Utviklingsmetodikk

Det har vært planlagt å bruke deler av Scrum for å bidra med struktur og oversikt. Oppgaver tilhørende prosjektet skulle bli samlet i en backlog og iterasjoner av utviklingen bli tildelt iterasjoner kalt sprinter. Oppgaver fra backlogen skulle fordeles på utviklerne på egne sprintmøter, og kortfattede

standup-møter ville bidra til jevn flyt i kommunikasjonen. Det viste seg at dette i praksis ble utfordrende å etterfølge. Dette beskrives nærmere i kapittel 7.1.

Dagene teamet jobber, starter med et kort standup-møte på maksimum 15 minutter. I dette møtet går hvert teammedlem gjennom det han har gjort siden forrige møte, og hva som skal gjøres frem til neste møte. Eventuelle problemer eller hindringer kan også adresseres på standup-møtet.

For å lære av brukererfaringer, vil det jobbes mot å ha et minimum fungerende produkt (MVP) allerede etter de to første iterasjonene. Det tas sikte på å besøke pilotkunden for brukertesting av MVP, men dette må avtales når teamet, i samråd med Adfectus, mener det er hensiktsmessig.

3.5.3 Risikovurdering

Det har blitt utarbeidet følgende risikoanalyse:

	Hendelse/risiko	Årsak	Sannsynlighet	Konsekvens	Risikoprodukt	Tiltak
1	Sykdom blant utviklere	Høy smitte i samfunnet	Middels (3)	Middels (3)	Middels (9)	God håndhygiene
2	Modulen virker ikke	Begrenset kunnskap om kodespråk og teknologi	Lav (2)	Høy (4)	Høy (8)	Tilstrekkelig egentrening og opplæring, brukertesting
3	Modulen brukes ikke	For vanskelig brukergrensesnitt, språk e.l.	Lav (2)	Høy (4)	Høy (8)	Brukertesting, tett dialog med kunde
4	Målet med modulen er ikke forstått	Mangel på dialog mellom utviklere og kunde	Svært lav (1)	Høy (4)	Høy (4)	Tett dialog med kunde
5	Tap av kode, eller korrupt kode	Ingen versjonskontroll, svake rutiner på git	Svært lav (1)	Svært høy (5)	Svært høy (5)	Gode rutiner i bruk av git

Figur 10: Risikoanalyse

Av disse risikoene har flere inntruffet, med varierende alvorlighetsgrad. Det har vært sykdom blant utviklerne, som har bidratt til å forsinke utviklingen av modulen noe. Det ble ikke tatt høyde for at sykdom blant barn av utviklerne også kunne påvirke progresjonen, noe som også inntraff i løpet av prosjektperioden. De gangene dette inntraff ble det aktuelle arbeidet fordelt på utviklere som ikke var rammet, slik at fremdriften til en viss grad ble opprettholdt.

Underveis i utviklingen virket ikke modulen som planlagt, men dette var en naturlig del av veien til målet, og hindret ikke videre utvikling.

3.6 Evalueringsplan

Måltidsplanleggeren skal evalueres ved å gjennomføre et antall tester som kategoriseres under *verifikasjon* og *validering*. Tester under verifikasjon skal sikre at løsningen er rett bygget, og tester i validering skal sikre at løsningen oppnår målsettingen (Helgesen, 2023).

Modulen skal testes så fort et minimumsprodukt er oppe og går. Det blir naturlig at Adfectus gjennomfører tester først, før modulen testes på barn i målgruppens alder. Tidspunkt for disse testene avhenger av når modulen er testklar.

3.6.1 Verifikasjon

3.6.1.1 Enhetstester og modultester

Enhetstester og modultester vil bli gjennomført fortløpende i prosjektperioden, for å sikre at komponentene i modulen fungerer etter intensjonen, og at modulen klarer å kjøre i det eksisterende miljøet. Det blir ikke lagt opp til utforming og gjennomføring av automatiserte enhetstester.

3.6.1.2 Integrasjonstest og systemtest

Disse testene gjennomføres også etter avsluttet iterasjon. I integrasjonstesten testes modulens integrering opp mot resten av appen, og at modulen samhandler korrekt med relaterte moduler. En systemtest avdekker om appen fungerer normalt etter at den nye modulen er sammenflettet med resten av appen.

3.6.1.3 Brukertest

Modulen skal testes på aktuelle brukere av modulen. Disse brukerne kommer til å være ansatte på Beitostølen og deres pasienter med foreldrene. Brukertester gjennomføres etter utvalgte iterasjoner er gjennomført, og minimumsproduktet er klart. Én eller to utviklere fra teamet bør være til stede under testing og observere modulen i bruk. Det gjøres nødvendige notater om hvordan testeren interagerer med modulen, og om testeren bruker modulen som forventet. Dersom fysisk observasjon ikke blir mulig, bør pilotkunden gjennomføre brukertest, med påfølgende intervju gjennomført av medlemmer av teamet.

3.6.2 Validering

Modulen gjennomgår akseptansetest i samarbeid med Adfectus, for å sikre at modulen oppfyller de krav som er formulert i kravspesifikasjonen. Det vil også være av stor betydning at pilotkunden mener modulen oppfyller sin hensikt. Akseptansetesten gjennomføres i avsluttende fase av prosjektet.

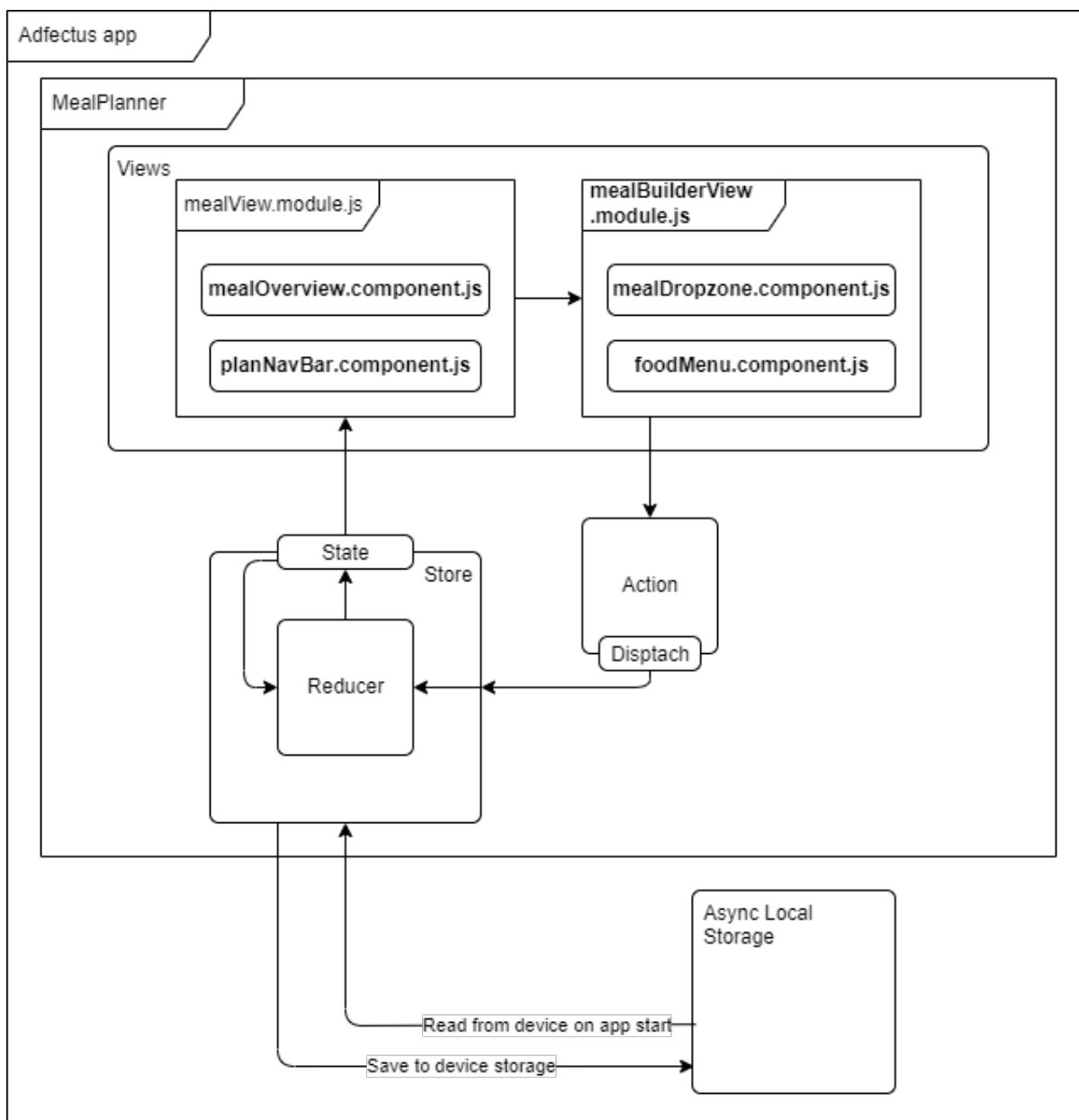
4 DETALJERT LØSNING

Måltidsplanleggeren er basert på en betaversjon av Adfectus-appen. Dermed er konfigurasjon av tilstandshåndtering, notifikasjoner, happtikk og navigasjon allerede utviklet. Dersom modulen trenger tilpasning av disse teknologiene, er det bygget på allerede eksisterende infrastruktur.

4.1 Inndeling av modulen

I prosjekter som involverer utvikling av programvare, er inndeling av moduler en viktig faktor som kan påvirke prosjektets suksess. I dette prosjektet har teamet jobbet med å finne den mest hensiktsmessige inndelingen av måltidsplanleggermodulen for å sikre enkelhet for unge barn. Selv om det ble valgt en løsning tidlig i prosjektet (se kapittel 3.3), har teamet fortsatt med diskusjoner og møter gjennom hele prosjektet, hvor løsningen har blitt kontinuerlig endret basert på oppdagelser gjort under utviklingen.

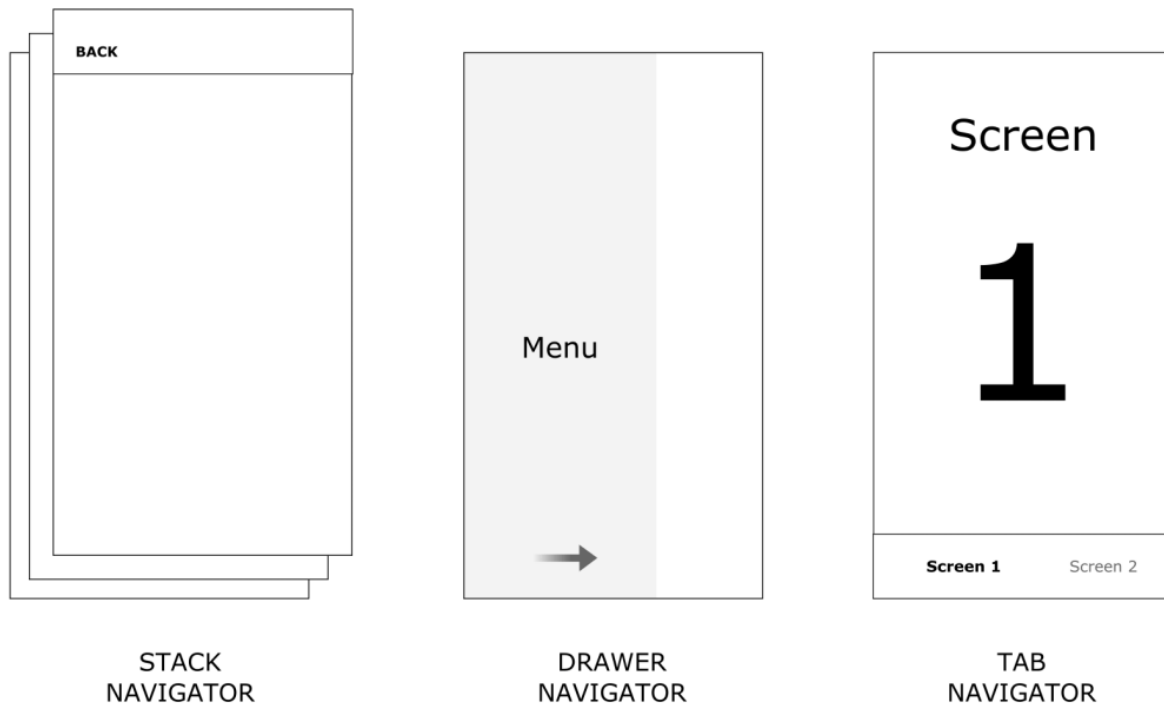
Den endelige inndelingen består av to views: En oversikt som viser de planlagte måltidene og en måltidsbygger. Figur 11 illustrerer hvordan modulen er inndelt og hvordan den er koblet opp mot biblioteket Redux for tilstandshåndtering (state management), som blir nærmere beskrevet i delkapittel 4.3 Lagring av data. Modulen har også fått navnet MealPlanner i koden ettersom all kode er skrevet på engelsk.



Figur 11: Arkitekturdiagram som viser flyten av data i modulen og hvordan komponentene er koblet sammen. Inspirasjon hentet fra: (Abramov, u.å., avsnitt 5).

4.1.1 Navigasjon

På nettsider brukes det en global stabel hvor sidene blir lagt til toppen («pushet») når brukeren trykker på en URL-kobling, og dermed legges sidene oppå hverandre. Nettleseren viser alltid den øverste siden på stabelen. Sammenlignet med nettleserapplikasjoner, har ikke React Native applikasjoner et innebyggt konsept for navigering mellom views. Derfor må utviklerne velge blant tre typer navigasjonsmuligheter som må implementeres gjennom biblioteker: Stack Navigator, Drawer Navigator og Tab Navigator, som illustrert i figuren under. (Navigation, u.å.).



Figur 12: Ulike navigasjonstyper for mobilapplikasjoner (Navigasjon, u.å.)

I Adfectus appen brukes det et Native Stack Navigator bibliotek for å navigere mellom og gjennom modulene. Fra startsidene, når en bruker trykker på en modul, blir den lagt på toppen av en stabel, og når de går tilbake blir den tatt av toppen («poppet»), på samme måte som med nettsider. Biblioteket kommer i to utgaver: Stack Navigator og Native Stack Navigator. Den viktigste forskjellen er at Native Stack Navigator bruker API-er som er laget for («native») iOS (UINavigationController) og Android (Fragment). På den måten vil den oppføre seg likt og ha samme ytelse som om navigasjonen var gjort i et utviklingsspråk som var skreddersydd for hvert operativsystem. Mens den andre er skrevet i JavaScript og kan derfor gi lavere ytelse (*Native Stack Navigator*, u.å.).

Tab Navigator er den mest vanlige navigasjonstypen i mobilapplikasjoner. En liten navigasjonsbar legges til på bunnen eller toppen av siden med knapper til andre views. Her legges et og et view i fokus av gangen. Bare viewet som er i bruk lastes inn («mounting»). (*Bottom Tabs Navigator*, u.å.). Denne typen passer best til stående applikasjoner, noe Adfectus appen ikke er.

Drawer Navigator gir muligheten for å dra inn en navigasjonsmeny på siden av skjermen, mens resten av innholdet blir skjøvet bak og nedtonet. (*Navigation*, u.å.). Den gir et elegant design, men er ikke nødvendigvis så intuitivt, særlig for mindre barn.

Utover navigasjon ble «overlays» vurdert. I stedet for at brukeren navigerer mellom views, blir nytt innhold dradd inn i viewet, og det gamle blir nedtonet på lik linje med Drawer Navigator. Men til forskjell er det ikke en navigasjonsmeny som blir dradd inn. Dette ble valgt bort fordi det ikke nødvendigvis ville være like intuitivt som valgte navigasjonstype, og det kunne gi en mer rottetete utseende.

4.1.2 Dagsoversikt

Brukeropplevelsen

Når matplanleggermodulen åpnes blir brukeren presentert med en oversikt over de neste måltidene. Disse vises som kort som enten er blanke om måltidet ikke er fylt ut, ellers inneholder de produktene som brukeren har valgt. Utseende til kortene er omtrent det samme som måltidsbyggeren, som beskrives i neste kapittel.

Kortene blir presentert ett av gangen der brukeren blar nedover for å vise neste kort. Det øverste kortet er neste måltid for den gjeldende dagen, og bare måltidene for den dagen vises. Alle planene har en datostempling som brukes for å sammenligne opp mot den gjeldende dagen. I dagsoversikten vil dermed forbigåtte måltider forsvinne fortløpende.

På høyre side kan ukedagene blås gjennom for å hoppe til ønskede dager. Denne er inndelt i uker for å øke oversikten. På samme måte som med måltidskortet, vil forbigåtte dager forsvinne. På bunnen av menyen vil det være en knapp for å opprette en ny uke.

I flere moduler bruker Adfectus avataren som brukeren lager, og normalt vil den reagere på valgene brukeren tar. På den måten vil appen føles mer tilpasset hver enkelt bruker, noe som er viktig for Adfectus. På den andre siden er avataren i måltidsplanleggeren passiv og reagerer ikke på det rundt den. Selv om den da ikke bidrar like mye, kan den fortsatt hjelpe litt på vei med tilpasset innhold gjennom at barnet kan se seg selv.



Figur 13: Dagsoversikten

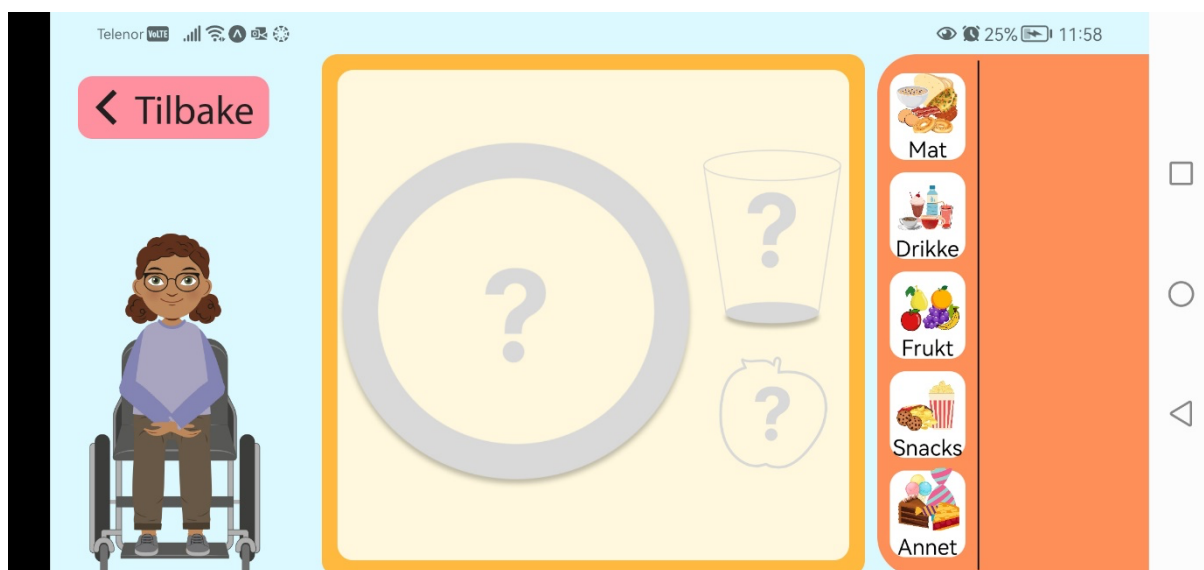
Teknisk implementasjon

Avataren er laget som en komponent og var derfor enkelt å sette inn i måltidsplanleggeren gjennom en import setning. Og ved å legge den til nå, vil det gjøre det lettere senere som en potensiell utvidelse at avataren reagerer på matvalgene brukeren tar. Sammen med en tilbakeknapp er de lagt i en egen komponent som gjør at de kan gjenbrukes lettere.

4.1.3 Måltidsbyggeren

Brukeropplevelsen

Fra dagsoversikten trykker brukeren på et kort og blir sendt videre til Måltidsbyggeren. Her setter brukeren sammen kommende måltider en av gangen.



Figur 14: Tomt måltid

På venstre siden vises avataren til bruken sammen med en tilbakeknapp til dagsoversikten. Mens til høyre er det en meny av mat- og drikkeobjekter. Og i midten vises måltidet som blir satt sammen.

I menyen velger brukeren først kategorien de har lyst å velge fra (mat, drikke, frukt, snacks eller annet). Herfra blir det vist en liste av alle mulighetene, som kan blas gjennom. Matproduktene representeres av bilder med tekst under. Brukeren vil da dra de ønskede objektene til midten av skjermen (droppsonen).

Droppsonen er delt i tre deler: Et fat for større matprodukter, et glass for drikke og et grått eple for resten. Selv om ulike produktene tilhører ulike soner, kan de bli dradd til hvor som helst på skjermen utenfor menyen til høyre, og så vil de automatisk bli plassert der de hører hjemme. Dette gjør modulen lettere å bruke, spesielt på mindre skjermer og for personer med nedsatt motorikk. Det er tenkt at brukeren skal kunne legge til én drikke, én frukt (eller annet tilbehør) og to større matprodukter.

Når brukeren trykker på et valgt matobjekt i droppsonen, så skal det komme opp en liten søppelbøtte i hjørnet av objektet for sletting. Dette er ikke implementert enda. Når en bruker har lagt til et objekt, er det derfor ikke mulig å fjerne den igjen direkte. Dette er ikke et stort problem ettersom objektene kan overskrives ved å dra nye tilsvarende objekter over.



Figur 15: Utfylt måltid

Når brukeren navigerer tilbake til dagsoversikten kommer måltidet opp i oversikten.



Figur 16: Utfylt måltid vist i oversikten

Teknisk implementasjon

Når brukeren trykker på et tomt kort for å lage et nytt måltid, blir det opprettet et nytt måltidsobjekt i JSON format:

```
meals: [{
  createdAt: Date,
  time: String,
  type: String,
  foods: [{
    id: Number,
    name: String,
    category: String,
    type: String,
    icon: SVG,
  }],
}]
```

Figur 17: JSON format for et måltid

Beskrivelse:

- time: Når måltidet er (f.eks. '08:00').
- type (meals): frokost, lunsj, middag eller kveldsmat.
- category: Skal brukes til å bedømme helsefaktorer. I gjeldende versjon av modulen er den alltid en tom streng og ikke i bruk enda.
- type (foods): mat, drikke, osv.

Denne blir sendt til Redux som endrer tilstanden til måltidsplanen, og deretter blir den hentet frem igjen i Måltidsbyggeren for så å kunne endres.

På venstre siden gjenbrukes tilbakeknapp- og avatarkomponenten, som nevnt ovenfor. Mens kjernen av dette viewet er delt i to komponenter: mealDropzone.component.js til venstre og food-menu.component.js til høyre.

At det er mulig å velge to større måltidsobjekter er tilfeldig valgt basert på hva som virket fornuftig for gruppen. Dette kan enkelt tilpasses etter testing og hva kundene ønsker. Alle måltidsobjekter brukt i prosjektet er midlertidig data, og dermed ikke bestemt etter hva som kommer til å ende opp i produksjon.

For lagring av planene var det først tenkt at når brukeren trykker «Tilbake»-knappen, skal det komme opp en melding om de ønsker å lagre. Dette ble forkastet til fordel at lagringen skjer hver gang innholdet i planen endres. På den måten vil ikke valg gå tapt selv om brukeren skulle lukke appen eller trykke feil. Ressursbruken blir større, men ikke på et vesentlig nivå slik at brukeren merker det.



Figur 18: Sletting av objekter. Søppelbotten dukker opp i det brukeren flytter på bamsen.

Sletting av matobjektene direkte ble prøvd implementert slik at brukeren kunne dra objektene fra midten av skjermen ut til siden og dermed forsvinne. Dette fikk ikke gruppen til å fungere, og dermed ble det diskutert alternative løsninger. Den ene var å dra til en søppelbøtte slik som i dørskilltlageren (se figur 18). Mens den andre var en sletteknapp på objektet når brukeren trykker på den. Dette er noe som kun ble diskutert, men ikke gjennomført som nevnt ovenfor.

4.2 Design og utforming av brukergrensesnitt

Modulen følger Adfectus-appens eksisterende visuelle uttrykk, slik intensjonen har vært fra prosjektteieren siden starten av prosjektet. Dette spiller en viktig rolle for å integrere modulen tilstrekkelig i appen, og for å styrke totalopplevelsen av appen. Farger som blir brukt, er hentet fra forhåndsdefinerte konstanter som importeres fra en fil i koden. Fargekoden for disse er bestemt av designeren for appen, og resulterer i en gjenkjennbar fargepalett.

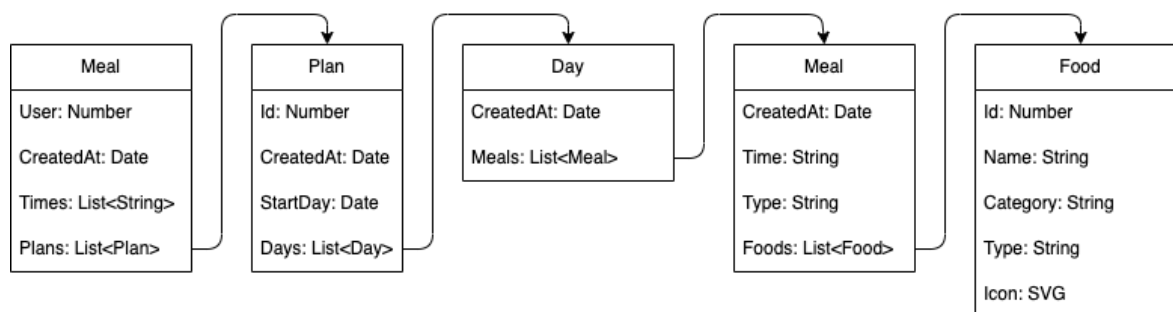
Bildefiler som representerer måltidsobjekter, som brød, melk, eple og så videre, er midlertidige, og er hentet fra internett. Disse blir erstattet med bilder designeren lager, når Adfectus måtte ønske å klargjøre modulen for lansering.

Brukergrensesnittet i modulen sikter på å balansere informasjonsmengde, slik at den aktuelle brukeren forstår hvordan de forskjellige delene henger sammen. Modulen har begrenset med tekstlige komponenter, og lener seg mer på grafiske elementer i forskjellige farger, som skal være selvforklarende for brukerne i målgruppen.

4.3 Lagring av data

Datastrukturen til modulen er hierarkisk. Den består av et måltidsobjektobjekt med generellinformasjon om bruker og en liste med alle planer. Et planobjekt inneholder informasjon om hver plan og en liste med alle dagene i uka. Et dagobjekt med ei liste av alle måltid for dagen. Et måltidsobjekt som inneholde informasjon om måltidet, og en liste av all mat og drikke i måltidet. Og til slutt et felles objekt for mat og drikke.

Data brukt i modulen skal lagres lokalt gjennom Redux og AsyncStorage.



Figur 19: Struktur av data som lagres og brukes i modulen.

Redux er et tilstandshåndteringssystem brukt for å håndtere data og dataflyt i modulen. Redux operer på en store(varehus) som holder alle tilstander er lagret på enheten. For å få tilgang til tilstander brukes det "hooks" gjennom "react-redux" biblioteket. Alle lagrede tilstander er uforanderlige, for å endre på en verdi må man lage en kopi av tilstanden, oppdatere kopien og deretter sende hele tilstanden tilbake.

For å endre tilstander brukes en "reducer", denne fungerer som et API. En "reducer" har "actions" disse er de forskjellige API-kallene man kan gjøre for å endre tilstander. Når man kaller en slik "action" henter "reduceren" en kopi av tilstanden før endring, endrer tilstand, og sender hele tilstanden tilbake varehuset. Varehuset lagrer så tilstanden lokalt på enheten igjennom AsyncStorage.

AsyncStorage er et vedvarende, asynkront, key-value lagringssystem for react-native applikasjoner. Dette blir brukt for å lagre tilstandene lokalt på enheten til bruker. Dette er implementert gjennom komponenter fra react-redux biblioteket.

4.4 Integrering med resten av appen

Måltidsplanleggeren får egen puslespillbrikke i hovedmenyen når den blir lansert i resten av appen. Dette designes og legges til av Adfectus i etterkant av prosjektets slutt.

Modulen gjenbraker en god del allerede eksisterende kode, men legger også til spesifikke metoder og datastruktur som bare måltidsplanleggeren tar i bruk. Det har blitt lagt til en måltidsreducer blant reducerne for appen, som håndterer aktivitet tilknyttet lokal lagring. Det har også blitt lagt til en datastruktur for måltider blant det som allerede finnes av datastrukturer i appen.

Modulen gjenbraker også andre deler av koden. Avataren som brukeren konfigurerer i en annen modul, dukker også opp i måltidsplanleggeren for å skape et gjenkjennbart element i brukergrensesnittet. Microservicer og konstantverdier blir også gjenbrukt. Sistnevnte inneholder også fargekodene for appen. Ved bruk av disse, begrenses fargebruken i brukergrensesnittet, slik at uttrykket i de forskjellige modulene samsvarer.



Figur 20: Hovedskjerm med puslespillbrikker. Måltidsplanleggeren vil få en egen puslespillbrikke på et egnet sted.

Relevante komponenter fra andre deler av appen blir også gjenbrukt. Et eksempel på dette er «Tilbake»-knappen (se figur 20), som både lagrer state, og går tilbake til det forrige viewet som ble vist.



Figur 21: Skjermbilde-sammensetning av andre moduler som bruker "Tilbake"-knappen (innringet).

Selv om måltidsplanleggeren har et lignende utseende og responderer på samme måte som andre deler av appen, må hele modulen gå gjennom en designprosess med den eksterne designer til slutt. Flere

deler av modulen inneholder stedfortredende bilder og farger, og vil mest sannsynlig bli justert etter prosjektets slutt. Det har vært presisert fra Adfectus' side at gruppen skal konsentrere seg mest om funksjonalitet og logikk, og ikke bruke for mye tid på design, nettopp av denne grunn.



Figur 22: Skjerm bilde av måltidsbyggeren. Midlertidige stedfortredere er brukt for matelementer, og skjermen inneholder minimalt med farger.

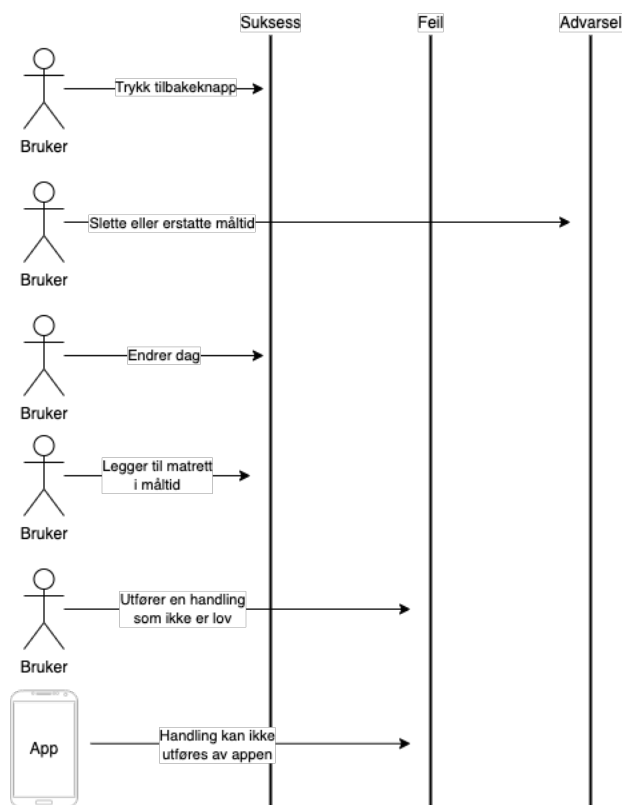
4.5 Tilrettelegging og Haptikk

Tilrettelegging og haptikk er utført etter retningslinjer utformet av Affectus og forskrift for universell utforming av ikt-løsninger utarbeidet av utilsynet. (Utilsynet, 2023)¹

Haptikk er allerede delvis utviklet av Affectus, men implementasjonen er ikke ferdigstilt. Prosjektet nyttet en egen implementasjon av haptikk gjennom Expo Haptics biblioteket.

```
onPress={()=> Haptics.notificationAsync(Haptics.NotificationFeedbackType.Success)}
```

Implementasjonen består av tre haptiske responser, suksess, feil og advarsel. Suksess signaliserer at handlingen bruker ønsker å utføre er vellykket, feil signaliserer at bruker mislykkes, og advarsel signaliserer at bruker må godkjenne handlingen for å fortsette.



Figur 23: Sekvensdiagram av handlinger som trigger en haptisk respons.

5 RESULTATER

Dette kapitlet omhandler hvordan prosjektresultater blir testet, evaluert, og hva resultatet ble. På grunn av manglende ressurser fra oppdragsgiver og utfordringer rundt tilstandshåndtering ble testperioden forkortet mye.

5.1 Evalueringsmetode

Resultater ble testet kontinuerlig igjennom prosjektet av gruppen og oppdragsgivers kontaktperson, Christer Kilavik. Når en MVP var ferdig utviklet ble den testet på andre medlemmer i Adfectus-teamet. Gruppen fikk tilgang til kildekoden for en beta-versjon av Adfectus-appen. Denne ble brukt for alt av utvikling til brukertest.

5.1.1 Enhetstest og modultest

For det meste av testing ble resultatet enten vist direkte på skjermen til en emulator eller skrevet ut i konsollen. Gruppen satt et mål om å bruke enhetstester for funksjonalitet og det ble gjort for noen funksjoner, men disse er ikke automatisert fordi modulen ble utviklet i et React Native-miljø og det medfører for mye arbeid for at gruppen kunne utføre det. Optimalt burde funksjonalitet som tilstandshåndtering og logikk testes i et eget miljø uavhengig modulen deretter integreres, men dette ble ikke utført fordi prosjektet skal videreutvikle allerede eksisterende systemer og komponenter.

Enkelte funksjoner, som dato-håndtering ble testet i et eget JS miljø uavhengig av utviklingsmiljøet. Tilrettelegging ble testet internt i teamet. For modulen testet vi flere fargekombinasjoner i egne miljø, deretter valgte vi en kombinasjon som kunne implementeres med Adfectus sine interne fargekoder.

5.1.2 Integrasjonstest og systemtest

Integrasjonstesting ble utført kontinuerlig igjennom hele prosjektet. Modulen er utviklet på toppen av en beta versjon av Adfectus-appen, dermed ble nye funksjoner og integrering med tilstandshåndtering testet direkte i Adfectus miljøet. Dette sikret at de er kompatible med resten av appen og trygt kunne innlemmes i appen.

Haptikk er implementert av Adfectus, men ikke tatt i bruk. Gruppen testet denne implementasjon internt på en beta versjon av modulen. En egen implementasjon ble utviklet, og testet internt.

5.1.3 Brukertest

Brukertesting ble gjennomført i slutten av prosjektet i samarbeid med Barneposten ved Haukeland Sykehus. Oppdragsgiver utpekte fire forskjellige mulige testgrupper: Adfectus-teamet, BHSS, familie og venner. Det to førstnevnte er interessenter for prosjektet. Gruppen vurderte å teste sammen med BHSS, men det ble ikke utført fordi prosjektet nådde aldri et punkt hvor vi vurderte at det var verd tiden. Senere i prosjektet hadde vært mer naturlig dersom modulen skulle testes på personer utenfor målgruppen. Et gruppemedlem har en sønn innenfor aldersgruppen, og det ble vurdert å teste modulen på han. Mot slutten av prosjekt perioden utførte vi tester på to barn, en gutt (7) og en jente (8).

Veileder anbefalte å unngå testing på personer i målgruppen før prosjektet nærmet seg et sluttprodukt, ettersom de har erfart at brukarene har problem med å forstå at produktet ikke er ferdig og dermed ikke gir verdifull tilbakemelding. Vi presiserer at testing på personer i målgruppen er svært viktig,

men ikke før endt prosjektperiode. Vi valgte allikevel å teste appen på personer i målgruppen, ettersom vi mener vi kunne hente verdifull læring om design og utforming.

Prosjektet ble frarådet å bruke BHSS for testing. Prosjektveileder mente, fra tidligere erfaring, at denne typen tester går over lengre tid og er derfor lite hensiktsmessig for dette prosjektet. Det er rett og slett ikke tid til å utføre en slik test. BHSS er lokalisert i Øystre Slidre kommune og derfor er det ikke lett for gruppen å teste der, ettersom gruppen holder til i Bergen.

For brukertesting nyttet prosjektet en beta versjon av modulen. Denne versjonen manglet noe funksjonalitet, som haptikk og tilrettelegging, logikk for visning av flere planer, og notifikasjoner. Gruppen ønsket å teste design, utforming, og brukervennlighet over ekstra funksjonalitet. Denne versjonen ble brukt for alle brukertester. Det ble utført interne og eksterne brukertester. De interne består av prosjektmedlemmene og de eksterne består av Adfectus-teamet og familie.

De eksterne brukertestene ble utført på to medlemmer i Adfectus-teamet og to barn. De ble presentert en tidlig versjon av modulen og fikk uavgrenset tilgang til å prøve hva som helst. Testperson nr.1 er et aktivt medlem i Adfectus-teamet og hadde derfor en del kunnskap om modulen, som kravspesifikasjoner og funksjonskrav. Testperson nr.2 er ikke et aktivt medlem av teamet og hadde derfor ingen forhåndskunnskaper om modulen. Han ble informert om det generelle formålet med modulen, men ikke noe mer. Testen er lik for barna. Under testen var de separert uten mulighet for kommunikasjon. Her ønsket gruppen å teste brukervennlighet og utforming.

5.2 Evalueringresultat

Under brukertesting observerte gruppen flere problemer som alle brukere opplevde. Disse inkluderte blant annet manglende mulighet til å slette valgte matretter og knapper som ikke fungerte. De fleste tilbakemeldingene fra brukerne under testingen var allerede blitt oppdaget av gruppen gjennom intern testing, og brukertesting bekreftet resultatene som var erfart internt.

Under brukertesten ble flere observasjoner gjort som gir et godt bilde av brukernes opplevelse. En av testpersonene hadde vanskeligheter med å forstå startskjermen, spesielt når det gjaldt dagsplanen. Imidlertid var det enkelt for testpersonene å håndtere og forstå "drag and drop"-funksjonaliteten som ble brukt i systemet. Det ble påpekt at noen matvarer manglet i systemet, og plasseringen av forskjellige matelementer var delvis forståelig, men det oppsto forvirring angående tilbehør og hva som kunne plasseres i eple-sonen.

Tilbake-knappen og kategoriinndelingen ble godt forstått av testpersonene, og de hadde en god oppfatning av designet i henhold til retningslinjene satt av oppdragsgiveren. Imidlertid var det en testperson som prøvde å fjerne et matelement fra en plan, men dette viste seg ikke å være mulig. Forståelsen av hvordan man kunne legge til en ny dagsplan var heller ikke helt klar for testpersonene.

På den positive siden ga testpersonene uttrykk for at systemet var generelt lett å forstå, tok ikke lang tid å lære seg, og opplevdes som brukervennlig. Selv om det oppsto noen utfordringer med "drag and drop"-funksjonaliteten, klarte testpersonene raskt å tilpasse seg og forsto hvor de måtte slippe maten for at den skulle legges til. Det ble også observert et gjentakende problem på iPhone-enheter, der testpersonene ved et uhell dro ned kontrollsenteret når de valgte matvarer.

Generelt sett er oppdragsgiveren fornøyd med resultatet. Under testingen påpekte de imidlertid at forsiden var noe uoversiktlig og forvirrende. De er derimot tilfredse med integrasjonen i Adfectus-appen. Design og utforming er utført i samsvar med retningslinjene som ble utarbeidet av oppdragsgiver. Oppdragsgiver mente imidlertid at strukturen for måltidene var forvirrende. Dersom

man ønsker å beholde det nåværende designet for måltidsbyggeren, må hver kunde legge inn sitt eget tilbud for mat og drikke for måltidsbyggeren.

Oppsummert viser brukertesten at det er noen områder som kan forbedres og forstås bedre av brukerne. Samtidig ble systemet generelt sett positivt vurdert, med enkel forståelse og rask læring av hvordan man bruker det.

Observasjoner fra brukertesting.

Resultat fra testperson nr.1:

- Man kan ikke fjerne matvarer dersom de er valgt, bare erstatte.
- Ønsket å legge inn matvarer fra tilbehør kategorien på feltet for hovedrett.
- Menyen med mat og drikke er forvirrende.
- Det var utydelig hva tallerken, glass og frukt symbolene betyr når man skal bygge måltid.
- Menyen på høyreside av forsiden gir ikke mening. Den burde bare vise en hel uke om gangen.
- Drag-and-drop er naturlig, men implementasjon fungerer ikke optimalt.

Resultat fra testperson nr.2:

- Når man drar en matrett fra menyen og inn på tallerken er det en lys bakgrunn som følger med.
- I dagsoversikten (feltet som viser planlagte måltid) er det ikke lett å se hvilket måltid som er relevant.
- Man kan ikke slette en matvare fra et måltid bare erstatte den. Dersom man har fire matretter i hovedmåltidet, kan man bare erstatte den siste.
- Ønsket en forklaring på hva man skal gjøre, eller en how-to-skjerm.

Resultat fra gutt (7):

- Ble forvirret av forsiden.
- Drag-and-drop er naturlig.
- Forsto at man bare kan erstatte matretter.
- Den var lett å forstå.

Resultat fra jente (8):

- Brukte litt tid før hun skjønnte drag-and-drop.
- Var tydelig hva som.
- Ønsket å fjerne en matvare fra planen, men det gikk ikke.
- Likte utseende og stil.

5.3 Prosjektresultat

På grunn av manglende ressurser hos Adfectus, har det blitt utfordrende å gjennomføre brukertester med representanter fra Adfectus, slik det opprinnelig var planlagt. Etersom brukertesting var mulig først mot slutten av prosjektperioden, vil resultatet fra testene ikke kunne påvirke prosjektresultatet, men bør likevel bli tatt hensyn til i fremtidig utvikling av måltidsplanleggeren.

Funn fra brukertester kan oppsummeres i følgende sentrale funn:

- Brukeropplæring gjennom en «how to»-skjerm vil være viktig for at brukerne av modulen skal komme raskt i gang med å bruke modulen effektivt og korrekt. Dette er et av delmålene i prosjektet, og vil være en naturlig del av arbeidet videre med modulen.
- Det er nyttig med forskjellige matkategorier, men at det må komme tydeligere frem hvor matelementer fra forskjellige kategorier kan plasseres i dropzone.
- Det å kunne fjerne valgt mat fra dropzone er en intuitiv operasjon og bør implementeres i fremtidige versjoner. Dette vil også oppfylle et av delmålene i prosjektet, ved å følge designet og interaksjonen til andre deler av appen. I dørschildlageren kan man dra grafiske elementer til et søppelspann for å fjerne elementet, og det vil være naturlig å bruke samme teknikken i måltidsplanleggeren også.
- Organisering av dager og uker i menyen på dagsoversikten må være tydelig og inndelt, og måltider bør ikke vises for flere dager, men kun for valgt dag. Dette vil også følge designprinsipper fra resten av Adfectus-appen. Informasjon på skjermen er tiltenkt barn, og må være tydelig og avmålt.

Selv om alle overnevnte punkt er områder for forbedring, har testene også vist at modulen har et godt utgangspunkt for videre arbeid. Barna som gjennomførte tester, som ikke har erfaring fra Adfectus-appen fra før, forstår modulens viktigste funksjoner. De klarte å komme i gang med å planlegge måltider etter kort tid, og plukket fort opp at drag and drop skulle brukes for å legge til mat i planene sine.

6 DISKUSJON

Prosjektet har resultert i en modul som danner et utgangspunkt for videre arbeid og utvikling. I dette kapitlet diskuteres prosjektets gjennomføring og hvordan resultatet har blitt påvirket av dette.

6.1 Prosjektgjennomføring

Omtrent hver dag gruppen jobbet på Eitri, ble det gjennomført statusmøter. Der ble det diskutert designvalg, progresjon og mål for tiden videre. Ekstern veileder var ofte med, og lot prosjektgruppen ta mesteparten av styringen på både møteform og valg knyttet til utformingen av modulen. Det var rom for både tilbakemelding og en anledning til å adressere problemer som hadde oppstått, eller stille spørsmål gruppen satt med. Etter fullført statusmøte fortsatte arbeidet ute i det åpne kontorlandskapet på Eitri, og oppgaver eller nødvendige justeringer som ble tatt opp på møtet, ble påbegynt.

Scrum som arbeidsmetode ble ikke gjennomført. Det viste seg å være en uvant måte å jobbe på, som krevde en god mengde disiplin for gruppen å komme ordentlig inn i, og det forsvant dermed i arbeidsmengden. Rullerende prosjektleder ble heller ikke gjennomført, men samarbeidet og kommunikasjonsflyten i gruppen holdt et såpass tilfredsstillende nivå, at dette viste seg å gå bra likevel.

Gantt-skjemaet og prosjekthåndboken som helhet ble ikke oppdatert så fortløpende som det opprinnelige ble lagt opp til. Dette arbeidet ble gjennomført i tilknytning til de obligatoriske innleveringene, og oppfylte i mindre grad sin hensikt. Timeføring hjalp til med å registrere og fange opp en arbeidsskjevhet underveis i prosjektet, noe som ble tatt opp i gruppen og jobbet videre med.

Iterasjoner på én og to uker viste seg å være utfordrende å gjennomføre i praksis, tatt i betraktning det arbeidet som måtte til for å produsere et MVP. Forsinkelser oppstod på grunn av utenforliggende faktorer, samt undervurdering av oppgaveomfang. I tillegg ble arbeidet med modulen jevnlig avbrutt av obligatoriske innleveringer av rapporter om arbeidet. Dette bidro til at det ble utfordrende å holde en jevn progresjon på kodingen. Løsningen ble at hvert gruppemedlem jobbet hvert med sitt, og ga hyppige oppdateringer over bordet og på digitale flater utenfor felles samarbeidstid. Rundt frister for obligatoriske innleveringer ble disse prioritert, noen ganger i strekk på 1-2 uker, og deretter ble koding gjenopptatt.

6.2 Påvirkninger på resultatet

En av de negative faktorene som påvirket resultatet, var alt for mye tid brukt på interne møter. Selv om møtene var nødvendige, hadde de en tendens til å drive av sporet, eller de gikk for mye i detaljer for hva som var nødvendig på det tidspunktet. For å forbedre dette, kunne møteinnkalleren ha laget en sakliste for å gjøre dem mer produktive – slik som ved møtene med intern veileder på HVL. På den måten ville møtene hatt en klarere struktur og lettere å holde strengt saklige. En annen mulighet hadde vært å sette opp en tidsramme for møtene som legger press på at det er litt fortgang.

Men på tross av dette var møtene nyttige for å diskutere og komme frem til løsninger som alle var enige om. Etter hvert møte satt alle igjen med en felles oppfatning av hva som skulle lages og hvordan det skulle arbeides med. Samtidig ble det gjort mange designendringer gjennom prosjektet, og etter hvert som noe ble gjort måtte neste steg diskuteres. Så derfor var det viktig med mange møter. Dette var en positiv faktor som påvirket resultatet av prosjektet.

En annen faktor som hadde negative innvirkninger på resultatet, var stadige innleveringer og obligatoriske krav i dette faget. De skapte avbrytelser i fremdriften ved at gruppen måtte skifte fokuset til å gjøre neste arbeidskrav, i stedet for å fortsette med planleggingen og utviklingen av prosjektet.

På den andre siden kan gruppen se utbytte av at faget er lagt opp slik. Analysen av tidligere oppgaver har vært nyttig for å øke nivået på skrivingen, samtidig som det gir et bra utgangspunkt for hvordan rapporten kan struktureres og hva den bør inneholde. Videre har innleveringene gjort at skrivingen på rapporten blir mer fordelt utover, noe som gjør den mer overkommelig og dermed kan bidra til å heve den. Det fører også til at veilederen kan bistå tidligere og oppdage eventuelle problemer så snart de oppstår.

Emneansvarlig anbefalte i en forelesning at gruppene burde delegere oppgavene ut ifra styrker, og at det var en fordel å ha ett medlem som har hovedansvaret for skrivingen av rapporten, mens resten jobber med utviklingen eller gjennomføringen av prosjektet. Dermed sikres det at alle områdene blir jobbet med kontinuerlig, uten større omskiftninger i arbeidsflyten. Dessverre var det ingen naturlig inndeling for denne gruppen, og selv om ett medlem fikk ansvaret for skrivingen, ble det gjennomført i mindre grad enn ønskelig.

Men når det gjaldt utviklingen av prosjektet, kom medlemmenes ulike styrker tydeligere frem. Her ble oppgaver som prototyping, arkitekturmodeller og lignende fordelt slik at én person tok seg av hver oppgave, der resten av teamet kom med innspill.

Ellers var det andre fag som opptok mer tid enn ønsket. Ettersom de i hovedsak bestod av obligatoriske innleveringer, kunne de ikke unnlates. Så det medførte at tid som helst skulle vært brukt til dette prosjektet, ble prioritert til de andre fagene.

Som på de fleste prosjekter ble dette også rammet av noe sykdom. Det er svært vanskelig å forhindre, og må derfor planlegges rundt. Som regel var ikke teamet avhengige at alle møtte opp hver gang for å kunne fortsette på prosjektet, og om det var noe som måtte diskuteres kunne den fraværende personen kontaktes gjennom Discord. Så med dette, førte ikke sykdom til større problemer enn at progresjonen ble minimalt redusert.

6.3 Analyse av resultat

Selv om det ikke har blitt gjort en omfattende brukertesting, kan prosjektets potensielle innvirkning være betydelig. Adfectus' visjon om å forenkle og effektivisere kommunikasjonen mellom barn og helsepersonell er et viktig bidrag til helsesektoren, som beskrevet i innledningen. Modulen laget i dette prosjektet er en naturlig utvidelse av deres arbeid, og vil hjelpe barn med å utvikle sunne matvaner fra en tidlig alder. Digitaliseringen av helsetjenester er en voksende trend, og Adfectus' arbeid tjener som et utmerket eksempel på hvordan teknologi kan utnyttes for å forbedre pasientresultater. I fremtiden vil utviklingen av lignende produkter for andre aldersgrupper og helseproblemer være avgjørende for å forbedre den generelle kvaliteten på helsetjenestene.

Selv om digitaliseringen er nødvendig, er det viktig å merke at den medfører noen risikoer. Særlig når det gjelder personer i sårbare posisjoner, som barn, er det høyst behørig å ta personvern alvorlig. Med et voksende behov for digitale verktøy, er det lett for å nedprioritere pasientenes sikkerhet og behov for menneskelige interaksjoner.

7 KONKLUSJON OG VIDERE ARBEID

I dette kapitlet presenteres et sammendrag av delmål og krav fra kapittel 1 og prosjektets visjonsdokument. Det blir gjennomgått i hvilken grad delmål og krav ble oppnådd, hva videre arbeid med modulen bør fokusere på, og til slutt hva andre kan lære av dette bachelorprosjektet.

7.1 Oppfylte delmål og krav

Det er blitt laget en måltidsbygger der brukere kan sette sammen et måltid fra en liste med mat- og drikkeelementer. Måltidenes innhold kan justeres i etterkant, etter brukerens behov.

Det er også blitt laget en oversikt over kommende måltid, slik at brukeren kan lage fullstendige dagsplaner, og få en visuell representasjon og samleoversikt over kommende måltider. Lagring av disse planene er implementert gjennom tilstandshåndtering og ved bruk av Redux.

Designet på modulen følger i stor grad designet som allerede er i bruk i Affectus-appen, og glir dermed godt inn blant allerede eksisterende moduler.

7.2 Delvis oppfylte delmål og krav

Kravet om gjenbruk av tidligere måltidsplaner er delvis oppfylt. Per nå kan planene kun lagres, men ikke slettes eller kopieres. Planene som er lagret, er ikke synlig for brukeren etter en spesifisert dato er passert. Alt dette er starten på veien mot funksjonalitet som lar brukeren gjenbruke planer.

Første gang modulen åpnes av bruker, skal det vises en «How to»-skjerm med opplæring i bruk av modulen. Alt tekstlig innhold til denne skjermen er skrevet, men den er ikke lagt inn slik at den vises slik intensjonen er. Grafikken til «How to»-skjermen mangler også, og må inkluderes når grafikk for hele modulen bestilles fra ekstern designer.

Avataren til brukeren er lagt til i modulen, men fungerer ikke tilstrekkelig interaktivt med de forskjellige operasjonene brukeren gjennomfører.

7.3 Hvorfor ikke alle mål ble oppfylt

Det er et sammensatt bilde av grunner til at ikke alle mål og krav ble oppfylt, som blant annet reflekteres i risikoanalysen (kapittel 3.5.3). Det totale grunnlaget til hvorfor ikke alle mål ble oppfylt, kan likevel kokes ned til to grunner: prioritering og balanse mellom rapportskrivning og koding, og teknologien som måtte læres mer eller mindre fra bunn.

7.3.1 Rapportskrivning vs. koding

Bachelorprosjektet var delt mellom rapportskrivning og koding, som utgjorde grunnlaget for rapportskrivningen. Før prosjektets start, uttrykte flere fagansvarlige ved HVL at studenter ofte ble overrasket over skrivemengden i bachelorprosjektet. I dette bachelorprosjektet erfartes det at rapportskrivningen tok vekk arbeidsflyt i kodingen. For å kjøpe fri arbeidstid til koding, ble det nedprioritert å holde prosjektdokumentasjon (f.eks. prosjekthåndbok) oppdatert jevnlig, noe som videre bidro til at det gikk med ekstra mye tid til å oppdatere denne før obligatoriske innleveringer.

7.3.2 Ny teknologi

React Native, og flere teknologier brukt i forbindelse med utviklingen, var nye for prosjektgruppen. Dette krevde at man jobbet seg opp fra et tilnærmet nullpunkt, til ganske kompleks funksjonalitet. Til tross for jevnlig oppfølging fra Adfectus, gjorde dette antageligvis at det ble brukt mer tid enn om det fantes mer kompetanse og erfaring internt i prosjektgruppen.

Det bør også nevnes i denne sammenheng at det ble brukt mye tid på versjonskontroll og utfordringer knyttet til blant annet konflikthåndtering i koden, sammenfletting av forskjellige utvikleres grener, og endret base for foreldregren.

7.4 Videre arbeid

Funksjonalitet som bør innlemmes i videre utvikling av modulen, bør inkludere

- gjenbruk av tidligere måltidsplaner
- brukerdefinerte innstillinger, eksempelvis der brukeren kan bestemme tidspunkt for måltid, eller legge til mellommåltid som ikke allerede finnes i dagsoppsettet
- tilrettelegging ved bruk av lydavspillinger og haptikk
- administratoroperasjoner, eksempelvis for å legge til spesifikke middager i brukeres måltidsplaner, eller å legge til mat- og drikkeelementer i matmenyene
- opplæring i ernæring, eksempelvis ved bruk av et poengsystem for ulike matvarers sunnheitsgrad, og variasjonslogikk som ivaretar at brukere ikke spiser det samme hver dag til hvert måltid

7.5 Lærdom andre kan ha av dette prosjektet

Prosjektgruppa har gjort seg mange erfaringer fra prosjektperioden. Dersom liknende prosjekt skal gjennomføres i fremtiden, kan tre lærdommer trekkes frem, alle i tilknytning til tidsbruk.

7.5.1 Opplæring i teknologi

Det bør settes av tilstrekkelig tid til egenøving og trening innenfor teknologien som skal brukes, inkludert versjonskontroll, om dette er nytt for gruppe medlemmene. Felles arbeidstid bør ikke brukes til å lære seg disse teknologiene, da forskjellige personer har bruk for trening i forskjellige teknologier. Tid man har satt av til felles arbeid spises fort opp dersom man bruker denne tiden til denne typen opplæring.

7.5.2 Struktur på møter

Erfaringen fra dette bachelorprosjektet var at de jevnlig statusmøtene på Eitri hadde en tendens til å flyte ut i både tidsomfang og tematikk. «Plutselig» hadde mye tid gått på å snakke om temaer og problemstillinger som ikke hadde en direkte innvirkning på prosjektarbeidet. Selv om dette vitner om god gruppekjemi, tjente det ikke prosjektet til slutt. Møter bør ha en tydelig struktur og agenda, og holdes korte og konsise.

7.5.3 Balanse mellom skriving og koding

Å finne en god balanse mellom rapportskriving og koding kan rimelig antas å være nøkkelen til et vellykket bachelorprosjekt. Hvor dette balansepunktet befinner seg varierer nok fra situasjon til situasjon. I dette bachelorprosjektet ble majoriteten av felles arbeidstid brukt på koding og relaterte problemstillinger, mens rapportskriving og arbeid på obligatoriske innleveringer ble gjort mer eller mindre individuelt. Skrivearbeidet hadde nok blitt mindre overveldende om dokumenter ble oppdatert fortløpende, og gjerne på faste, avsatte tidspunkt.

8 REFERANSER

- Abramov, D. (u.å.) *Redux Fundamentals, Part 2: Concepts and Data Flow*. Tilgjengelig fra: <https://redux.js.org/tutorials/fundamentals/part-2-concepts-data-flow> (Hentet 28.04.2023).
- Adfectus (2022) *Om Oss*. Tilgjengelig fra: <https://www.adfectus.no/om-oss> (Hentet: 23. februar 2023).
- Aziz, N. A. A. (2013) Children's Interaction with Tablet Applications: Gestures and Interface Design, *International Journal of Computer and Information Technology*, 02 (03), s. 447-450. Tilgjengelig fra: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=bc14015e89152a54e9d6bc42bc2230188e9a3805> (Hentet: 24. februar 2023).
- Beitostølen Helse- og senter (BHSS) (2023) *Om Beitostølen Helse- og senter*. Tilgjengelig fra: <https://www.bhss.no/om-senteret/> (Hentet: 23. februar 2023).
- Bottom Tabs Navigator* (u.å.) Tilgjengelig fra: <https://reactnavigation.org/docs/bottom-tab-navigator> (Hentet: 07.05.2023).
- Helgesen, C. 2023. DAT191 Bacheloroppgaven – Rapportskrivning. *DAT191-1 22H Bacheloroppgave*. Tilgjengelig fra: https://hvl.instructure.com/files/2182554/download?download_frd=1 (Hentet: 21. februar 2023).
- Native Stack Navigator* (u.å.) Tilgjengelig fra: <https://reactnavigation.org/docs/native-stack-navigator> (Hentet 29.04.2023).
- Navigation* (u.å.) Tilgjengelig fra: <https://react-native-course.elazizi.com/navigation> (Hentet: 07. mai 2023).
- Navigation* (u.å.) [Digital illustrasjon]. Tilgjengelig fra: <https://react-native-course.elazizi.com/navigation> (Hentet: 07. mai 2023).
- NOU 2023: 4 (2023) *Tid for handling: Personellet i en bærekraftig helse- og omsorgstjeneste*. Oslo: Departementenes sikkerhets- og serviceorganisasjon, Teknisk redaksjon.
- Paul Joachim Bloch Thorsen* (u.å.) Tilgjengelig fra: <https://ehin.no/2021/speakers/paul-joachim-bloch-thorsen/> (Hentet 18. mai 2023).
- Pedersen, R.M. (2021) *Samproduksjon i psykisk helsevern for barn og unge: En studie av pasientsentrering gjennom digitale helseinnovasjoner*. Masteroppgave, Høgskulen på Vestlandet.
- Syslak, A. I. (2019) *Alias, Designing an application for creating personalized comics aimed for the Children and Youth Clinic at Haukeland University Hospital*. Masteroppgave, Høgskulen på Vestlandet, Universitetet i Bergen. (Upublisert).
- Techopedia (2023) *What Does Drag And Drop Mean?* Tilgjengelig fra: <https://www.techopedia.com/definition/6918/drag-and-drop> (Hentet: 8. mars 2023).
- Thorsen P. (2019) *Safe and Better Communications with Personalized Visual Information - Making information understandable, motivating and engaging!* (Hentet: 21. februar 2023).
- Tilsynet for Universell Utforming av ikt (2023) *Universell utforming av apper*. Tilgjengelig fra: https://www.uutilsynet.no/regelverk/universell-utforming-av-apper/230#kravene_for_apper_er_wcag_20 (Hentet 18. mai 2023).

9 VEDLEGG

Vedlegg 1: Gantt-diagram

Vedlegg 2: Retningslinjer for kode og versjonskontroll

Vedlegg 3: Prosjekthåndbok

Vedlegg 4: Visjonsdokument

Vedlegg 5: Kravdokument

Vedlegg 6: Systemdokumentasjon

Vedlegg 7: Kildekode