

Oline Amundsen  
Ola Kristoffer Hoff

# MVP for mobilapplikasjon med algoritme som hjelpemiddel for type 1 diabetikere

Bacheloroppgave i Dataingeniør  
Veileder: Elise Klæbo Vonstad  
Mai 2022



Oline Amundsen  
Ola Kristoffer Hoff

# **MVP for mobilapplikasjon med algoritme som hjelpemiddel for type 1 diabetikere**

Bacheloroppgave i Dataingeniør  
Veileder: Elise Klæbo Vonstad  
Mai 2022

Norges teknisk-naturvitenskapelige universitet  
Fakultet for informasjonsteknologi og elektroteknikk  
Institutt for datateknologi og informatikk



Kunnskap for en bedre verden



---

## Sammendrag

Diabetes type 1 er en alvorlig og tidskrevende sykdom, som kan medføre store konsekvenser dersom den blir dårlig regulert. For å unngå komplikasjoner ved sykdommen, er det svært viktig at diabetikere klarer å regulere blodsukkeret på en god måte. Det er en sammenheng mellom hvor mye tid en diabetiker bruker på å regulere blodsukker og hvor godt regulert de er. Målet var å minke tiden brukt og forbedre reguleringen. Oppgaven har omhandlet diabetes type 1, og blir referert til som «diabetes» heretter, med mindre det blir spesifisert noe annet.

I denne oppgaven har det blitt utviklet en MVP (til en mobilapplikasjon) med en algoritme som skal beregne korrekt insulinmengde til måltider og korrigeringsinsulin basert på tidligere loggført data. Det har vært et fokus både på utviklingen og implementasjonen av algoritmen, samt systemutviklingen av MVP-en, da spesielt med fokus på et intuitivt design for å redusere tiden brukt på reguleringen av blodsukkeret. I tillegg har det vært et fokus på å analysere resultatene av algoritmen.

MVP-en er utviklet med Xamarin Forms, og den lagrer loggført data i en lokal database. Algoritmen bruker denne dataen til å gjøre endringer i skalarer til ulike matvarer, dagsprofiler og korrigeringsinsulin, og foreslår en insulinmengde til måltider og korrigeringsinsulin av blodsukker.

De preliminnære resultatene fra oppgaven viste at det hverken var en forbedring eller forverring i reguleringen av blodsukkeret til testpersonen. De viste derimot at tiden brukt på å regulere blodsukkeret gjennomsnittlig ble reduserte med rundt 40% i uken.

---

## Abstract

Diabetes type 1 is a severe and time consuming illness which can have dire consequences if not properly taken care of. It is of extreme importance that the blood sugar of a diabetic is properly regulated, in order to avoid complications related to the illness. There is a significant relationship between the time used to regulate the blood sugar, and how well regulated a diabetic is. The goal was to lower the time used to regulate the blood sugar, and improve the regulation.

In this bachelor-thesis there has been developed a MVP (to a mobile application) with an algorithm that will calculate a correct insulin dosage to meals and correction of the blood sugar, based on previously logged data. There has been a focus of the development and implementation of the algorithm, as well as the system development of the MVP, particularly on the intuitive design in order to reduce the time used to regulate the blood sugar. In addition, there has been a focus on analysing the results of the algorithm.

The MVP is developed with Xamarin Forms, and it stores data in a local database. The algorithm uses this data to make changes to the scalars for groceries, dayprofiles, and correction insulin, and it suggests an insulin dosage for meals and correction of the blood sugar.

The preliminary results show that there was not an improvement nor a deterioration in the regulation of the blood sugar of the testperson. However, the preliminary results showed that the time spent on regulating the blood sugar, on average, was reduced about 40% per week.

---

## Forord

Denne bacheloroppgaven har blitt gjennomført av Ola Kristoffer Hoff og Oline Amundsen, to dataingeniørstudenter ved NTNU, våren 2022. Oppgaven er gitt av Ola Kristoffer Hoff, som ønsket å undersøke i hvilken grad en algoritme og en applikasjon i form av en MVP, kunne forbedre reguleringen av blodsukkeret til diabetikere, samt tiden en diabetiker kan spare ved å ta i bruk MVP-en.

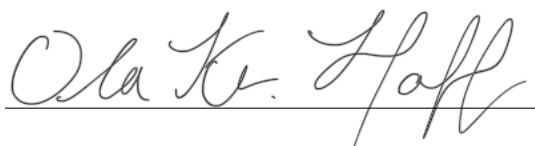
Studentene utarbeidet oppgaven selv. Oppgaven ble valgt da den ene studenten er diabetiker, og hadde et ønske om å utvikle et slikt verktøy. I tillegg åpnet oppgaven for å hjelpe med et eksisterende problem som begge studentene anser som svært viktig, da algoritmen kan bedre folks helse.

Prosesen har gitt studentene gode kunnskaper og erfaringer fra et vitenskapelig, teknologisk, og ingeniørfaglig perspektiv. Å utarbeide en plan og en MVP, samt å diskutere fremgangsmetoder og vitenskapelige metoder har vært utfordrende. Det har vært krevende å diskutere og konkludere resultatene, men den ferdigstilte MVP-en og konklusjonen av resultatene, er begge gruppens medlemmer fornøyde med.

Studentene ønsker å takke veileder for god hjelp og veiledning gjennom prosessen. I tillegg ønsker studentene å takke Janne Cathrin Hetle Aspheim, Hilde Hefte Haug, og Ole-Magnus Høiback fra «Statistikkhjelpen for bacheloroppgave i ingeniørfag» for god rådgøring rundt datainnsamling.



Oline Amundsen  
Trondheim, 19.05.2022



Ola Kristoffer Hoff  
Trondheim, 19.05.2022

---

## Oppgavetekst

Den opprinnelige oppgaveteksten ved prosjektets start var:

### **Hensikten med oppgaven:**

Hensikten med oppgaven er å lage en teknologisk løsning, i form av en app, som skal hjelpe diabetikere med å holde oversikt over hvilke matvarer de spiser og hvordan de skal regulere blodsukkeret. Mange diabetikere, spesielt ungdommer, har dårligere/mindre kontroll på reguleringen av blodsukkeret og dette kan medføre senskader og andre helseproblemer. Tanken er at appen skal gjøre det betraktelig lettere for diabetikere å ta vare på blodsukkeret på best mulig måte, med minimal innsatts fra sluttbruker. Målet er ikke å lansere appen, men heller å lage et "proof of concept" som eventuelt kan tas videre i samarbeid med helsepersonell.

### **Kort beskrivelse av oppgaveforslag:**

Foreslått løsning er en app der sluttbruker lett kan legge inn hva det nåværende blodsukkeret deres er, og hva (slags matvarer) de skal spise. Utifra dette vil appen håndtere alt det relevante av utregningen av insulin, som tar hensyn til f.eks. tidspunktet på dagen, om tidligere insulin fortsatt er aktivt, om personen nylig har trent ol.. Sluttbruker vil få en foreslått mengde insulin med minimal innsatts, og vil sette en mer presis dose insulin enn de ellers ville gjort. Da det er styrete å ta hensyn til alle relevante faktorer, brukes den "tradisjonelle" metoden da den er lettvin og enkel å forholde seg til for alle brukere (se utfyllende beskrivelse). Målet med appen er å gjøre det like lettvin og enkelt å forholde seg til, men å ta med flere relevante faktorer for et bedre regulert blodsukker. Brukerene trenger heller ikke endre på de foreslåtte profilene (se utfyllende kommentar) selv, håpet er at dette kan gjøres sammen med sykepleiere på brukerens kontroller hos sykehuset.

### **Utfyllende kommentarer til hva oppgaven gjelder:**

Oppgaven vil trenge utvikling av en MVP av appen, til både iOS og Android (hovedfokus på iOS), gjennom bruken av «Xamarin forms» og C# for UI, og C# for backend. Den "tradisjonelle" metoden går ut på at man regner ut en mengde insulin basert på nåværende blodsukker, ønsket blodsukker og antall karbohydrater i maten, basert på en fast "sensitivitet" (ofte regnet ut sammen med sykepleier/lege) til insulinet. Deretter vil man rette opp for at man eventuelt trenger mindre insulin om man har trent, eller om man fortsatt har insulin aktivt i kroppen. Disse faktorene kan ofte bli påvirket av magefølelsen, mens appen skal huske hva den har foreslått tidligere og gjøre det lettere å justere disse verdiene slik at de blir mer presise og korrekte.

### **Minimums funksjonalitet:**

- En «kalkulator» som tar verdien av blodsukker og antall karbohydrater som input, og gir en mengde insulin som output.
- Appen bruker innlagte «tids-profil»-er som f.eks. «frokost», «middag», ol. som har forskjellige «sensitiviteter» for insulinet. Riktig «tids-profil» vil bli valgt av appen basert på tidspunkt på dagen, men bruker skal manuelt kunne endre den. "Tids-profil"-er blir brukt da tidspunktet på dagen har innvirkning på hvor sensitiv kroppen er på insulin.

(fortsetter på neste side.)



- 
- Bruker skal kunne få opp en liste med matvarer som brukeren lett skal kunne plukke fra og velge antall å legge til. F.eks. use-case-en «jeg skal spise tre brøds-kiver og et glass appelsinjuice». Appen skal da legge inn antallet karbohydrater dette tilsvarer automatisk. Det trengs da en database over alle matvarene bruker legger til, og det er ønskelig og kanskje sette opp en standard «start» database med en del matvarer allerede lagt inn. Et innlegg i databasen trenger ikke nød-vendigvis å være en matvare, men kan også være en kombinasjon av matvarer, som f.eks. to brøds-kiver med syltetøy og et glass melk. Hvis bruker pleier å spise denne «matretten» til hver frokost, kan dette legges til som en «matrett», og bruker vil dermed slippe å måtte legge til hver enkelt matvare hver dag.

#### **Videre utvikling ønsket:**

- Bruker kan søke opp matvare basert på bilde av strekkoden til produktet.
- Bruker skal frivillig kunne legge inn «events» som f.eks. «lav blodsukker», «tren-ing», ol. Appen vil bruke denne statistikken til å påpeke trender som kan være vanskelig for en bruker å legge merke til selv. F.eks. «8 av de 10 siste gangene du har spist ris så har blodsukkeret blitt lavt 1-2 timer etterpå, vil du justere sensitiviteten din til ris?».
- Det hadde vært interessant om appen kunne knyttes opp mot sensordataene som enkelte av brukere har, å anvende disse datene i statistikken. Sensordataene kommer fra sensorer som sitter på kroppen og kontinuerlig måler blodsukkeret.
- Man kan skru på vasler slik at appen spør hvordan blodsukkeret ble etter X-antall timer, dette blir en «event» som vil hjelpe statistikken og kreve veldig lite innsats av bruker.
- Det er ønskelig å kunne lage forskjellige «hovedprofiler», altså forskjellige sett med «tids-profil»-er. Dette er nyttig da mange opplever at insulin behovene deres er annerledes i vanlig hverdag, helger, ferier, ved sykdom, syklus til kvinner, ol. Man kan da sette opp at appen bytter automatisk til f.eks «helg»-profilen til en bruker på lørdag kl. 00:01. og tilbake til «hverdag»-profilen på mandag. En bruker kan selvfølgelig også bytte mellom disse profilene manuelt etter behov. Oppgaven er relativt åpen utover minimums kravene. Det hadde vært ønskelig å kunne utvikle den videre basert på brukertester og den ønskede videre utviklingen.

Etter diskusjon innad i gruppen, ble det bestemt at minimumsfunksjonaliteten som omhandlet «matretter», skulle sløyfes, da dette ikke ville bidra til å forbedre algoritmen i noen stor grad. Det ville heller ikke bidra til å forklare resultatene i analysen bedre, enn det enkelte matvarer allerede gjorde. Etter bruk av MVP-en uttrykte ikke testpersonen et behovet for «matretter» heller. I tillegg innså gruppen at funksjonaliteten «påminnelser», punkt fire i listen over «Videre utvikling ønsket», var essensiell for algoritmens evne til å beregne og lære av gitte insulinmengder. Dette blir diskutert i delseksjon 5.4. Denne funksjonaliteten ble dermed prioritert opp til minimumsfunksjonalitet. Funksjonaliteten blir beskrevet i mer detalj i krav- og visjonsdokument, se vedlegg D.

---

# Innholdsfortegnelse

<b>Sammendrag</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Forord</b>	<b>iii</b>
<b>Oppgavetekst</b>	<b>iv</b>
<b>Figurliste</b>	<b>5</b>
<b>Tabelliste</b>	<b>6</b>
<b>1 Introduksjon og relevans</b>	<b>9</b>
1.1 Bakgrunn for behov . . . . .	9
1.2 Problemstilling . . . . .	9
1.3 Rapportens struktur . . . . .	9
<b>2 Teori og relevant litteratur</b>	<b>11</b>
2.1 Diabetes type 1 . . . . .	11
2.1.1 Insulin . . . . .	13
2.2 Statistikk . . . . .	13
2.2.1 Enkel lineær regresjon . . . . .	13
2.2.2 Regresjonsmodellen . . . . .	13
2.2.3 Regresjonslinjen og minste kvadratersmetode . . . . .	13
2.2.4 Varians . . . . .	14
2.2.5 Standardavvik for lineær regresjon . . . . .	14
2.2.6 Regresjonsmodellens godhet . . . . .	14
2.2.7 Forventning . . . . .	15
2.2.8 Konfidensintervall for ukjent linje . . . . .	15
2.2.9 Prediksjonsintervall for observasjonene . . . . .	15
2.2.10 Kovarians og korrelasjon . . . . .	16
2.2.11 Interrupted time series . . . . .	16
2.2.12 Root mean squared error . . . . .	17
2.3 Algoritme konstruksjon . . . . .	17

---

2.4	Systemutviklingsmetodikk . . . . .	18
2.5	Brukertester . . . . .	18
2.6	Sensor, LibreLink . . . . .	19
2.7	Tidligere relevant arbeid . . . . .	19
<b>3</b>	<b>Metode</b>	<b>20</b>
3.1	Forskningsmetode . . . . .	20
3.2	Prosess . . . . .	21
3.2.1	Informasjonsinnhentning . . . . .	21
3.2.2	Testgruppe . . . . .	21
3.2.3	Datainnsamling . . . . .	22
3.2.4	Databehandling . . . . .	23
3.2.5	Sikkerhetsmekanismer . . . . .	23
3.2.6	Testing . . . . .	24
3.2.7	Presisjon . . . . .	24
3.2.8	Tidsbruk . . . . .	25
3.3	Utførelse . . . . .	25
3.3.1	Eksperiment prosess . . . . .	25
3.3.1.1	Testperiode . . . . .	25
3.3.1.2	Datainnsamling . . . . .	27
3.3.1.3	Parametere til algoritmen . . . . .	27
3.3.2	Resultatanalysen . . . . .	28
3.4	Teknologi . . . . .	28
3.4.1	Frontend-rammeverk . . . . .	28
3.4.2	Backend . . . . .	29
3.4.2.1	Databaser . . . . .	29
3.5	Arbeids- og rollefordeling . . . . .	29
3.5.1	Parprogrammering . . . . .	30
3.6	Statistikk . . . . .	30
3.6.1	Interrupted time series . . . . .	30
3.7	Algoritmen . . . . .	31
3.7.1	Konstruksjon . . . . .	31
3.7.2	Virkemåte . . . . .	33

---

3.7.3 Alternativ . . . . .	38
3.8 Brukertester . . . . .	39
3.9 Måloppnåelse . . . . .	40
3.9.1 Resultatmål . . . . .	40
3.9.2 Effektmål . . . . .	40
3.10 Systemutvikling . . . . .	41
3.10.1 Utvikling . . . . .	41
3.10.2 Metodikk . . . . .	41
<b>4 Resultater</b>	<b>43</b>
4.1 Vitenskapelige resultater . . . . .	43
4.1.1 Algoritmen . . . . .	43
4.1.1.1 Matvarer . . . . .	43
4.1.1.2 Dagsprofiler . . . . .	48
4.1.1.3 Korrigeringsinsulin . . . . .	58
4.1.1.4 LibreLink . . . . .	61
4.1.1.5 Korrelasjonsmatriser . . . . .	63
4.2 Ingeniørfaglige resultater . . . . .	64
4.2.1 Resultatmål . . . . .	64
4.2.1.1 Tidsbruk . . . . .	65
4.2.1.2 HbA1c . . . . .	65
4.2.2 Effektmål . . . . .	66
4.2.3 Frontend . . . . .	66
4.2.4 Ikke-funksjonelle egenskaper og andre krav . . . . .	67
4.2.5 Kravdokumentasjon . . . . .	67
4.2.6 Brukertester . . . . .	67
4.3 Administrative resultater . . . . .	68
4.3.1 Fremdriftsplan og måloppfyllelse . . . . .	68
4.3.2 Planlegging og tidsbruk . . . . .	69
4.3.3 Systemutvikling . . . . .	69
<b>5 Diskusjon</b>	<b>71</b>
5.1 Dataintegritet og datagrunnlag . . . . .	71

---

---

5.2	Algoritmen . . . . .	72
5.2.1	Konstruksjon . . . . .	72
5.2.1.1	Partisjonerings metode . . . . .	72
5.2.1.2	Insulin til glukose forholdet . . . . .	73
5.2.1.3	Overlappende måltider . . . . .	74
5.2.1.4	Skalaren for korrigeringsinsulin uten matvarer . . . . .	75
5.2.2	Presisjon . . . . .	75
5.2.3	Parametere til algoritmen . . . . .	75
5.2.3.1	Minste antall datapunkter . . . . .	75
5.2.3.2	Nedre grense for prediksjonsintervall . . . . .	76
5.2.3.3	Absolutt største justeringsfaktor . . . . .	76
5.2.4	Insulinets virketid . . . . .	76
5.3	Oppnådde mål . . . . .	77
5.3.1	Korrelasjonsmatriser . . . . .	77
5.3.1.1	Gjennomsnittlig glukose . . . . .	77
5.3.1.2	HbA1c-estimat - GMI . . . . .	77
5.3.1.3	Glukose variasjon . . . . .	77
5.3.1.4	Veldig høye glukoseverdier . . . . .	78
5.3.1.5	Høye glukoseverdier . . . . .	78
5.3.1.6	Ønskede glukoseverdier . . . . .	78
5.3.1.7	Lave glukoseverdier . . . . .	79
5.3.1.8	Veldig lave glukoseverdier . . . . .	79
5.3.2	Algoritmen . . . . .	79
5.3.3	Resultat- og effektmål . . . . .	82
5.3.4	Ikke-funksjonelle egenskaper og krav . . . . .	84
5.4	Funksjonalitet i frontend . . . . .	84
5.5	Administrativt . . . . .	85
5.6	Samarbeid . . . . .	86
5.7	Systemutvikling . . . . .	87
5.8	Feilkilder . . . . .	89
5.8.1	Bias . . . . .	89
5.8.2	HbA1c-verdi . . . . .	89

---

5.8.3 Trening . . . . .	89
5.8.4 Mangel på data . . . . .	89
5.8.5 «Insulin-til-glukose»-forholdet . . . . .	90
<b>6 Konklusjon og videre arbeid</b>	<b>91</b>
6.1 Konklusjon . . . . .	91
6.2 Videre arbeid . . . . .	91
<b>7 Samfunnspåvirkning</b>	<b>93</b>
7.1 Etikk . . . . .	93
7.2 Vitenskap . . . . .	95
7.2.1 Prosess . . . . .	95
7.2.2 Resultater . . . . .	96
7.3 Bærekraft . . . . .	96
7.3.1 Miljø . . . . .	97
7.3.2 Samfunn og helse . . . . .	97
7.3.2.1 FNs bærekraftsmål . . . . .	98
7.3.3 Økonomi . . . . .	98
<b>Referanser</b>	<b>100</b>
<b>Vedlegg</b>	<b>104</b>
A Forskningsmetode . . . . .	104
B Algoritmen visualisert . . . . .	105
C Livsløpsanalyse og beregninger . . . . .	105
D Vedlagte dokumenter i prosjektmappe . . . . .	106
E Issue-board . . . . .	106
F Brukertester . . . . .	107
F.1 Brukertest 1 - Ikke diabetiker . . . . .	107
F.2 Brukertest 2 - Diabetiker . . . . .	111
<b>Figurliste</b>	
1 Total data på brød . . . . .	43
2 Utviklingen av matvaren «brød» . . . . .	44

---

---

2	Utviklingen av matvaren «brød» fortsettelse . . . . .	45
3	Utviklingen av matvaren «yoghurt» . . . . .	46
4	Utviklingen av matvaren «appelsinjuice» . . . . .	47
5	Utviklingen av matvaren «bringebærsyltetøy» . . . . .	48
6	Utviklingen av CarbScalar ved frokost . . . . .	49
7	Utviklingen av GlucoseScalar ved frokost . . . . .	50
8	Utviklingen av CarbScalar ved lunsj . . . . .	51
9	Utviklingen av GlucoseScalar ved lunsj . . . . .	52
9	Utviklingen av GlucoseScalar ved lunsj . . . . .	53
10	Utviklingen av CarbScalar ved middag . . . . .	54
11	Utviklingen av GlucoseScalar ved middag . . . . .	55
12	Utviklingen av CarbScalar ved kveldsmat . . . . .	56
13	Utviklingen av GlucoseScalar ved kveldsmat . . . . .	57
14	Utviklingen av korrigeringsinsulin . . . . .	58
14	Utviklingen av korrigeringsinsulin fortsettelse . . . . .	59
14	Utviklingen av korrigeringsinsulin fortsettelse . . . . .	60
15	Tid i ulike glukosesoner . . . . .	61
15	Tid i ulike glukosesoner fortsettelse . . . . .	61
16	Glukosehåndtering . . . . .	62
17	Forskningsmetode . . . . .	104
18	Algoritmen visualisert . . . . .	105
19	Issue-board . . . . .	106

## Tabelliste

1	Korrelasjonsmatrise . . . . .	63
2	Korrelasjonsmatrise før testperiode . . . . .	63
3	Korrelasjonsmatrise under testperiode . . . . .	64
4	Tabell over tidsbruk i ulike uker før MVP-en . . . . .	65
5	Tabell over tidsbruk i ulike uker med MVP-en . . . . .	65

---

## Akronymer

<b>TDD</b>	Total Daily Dose
<b>ITS</b>	Interrupted times series
<b>MVP</b>	Minimum viable product
<b>MVVM</b>	Model-view-viewmodel
<b>FN</b>	Forente nasjoner
<b>NSPE</b>	National society of professional engineers, Norsk: Nasjonalt samfunn av profesjonelle ingeniører
<b>NITO</b>	Norges ingeniør- og teknologorganisasjon
<b>Tekna</b>	Teknisk-naturvitenskapelig forening
<b>mAh</b>	Milliamperetimer
<b>Wh</b>	Watttime
<b>kWh</b>	Kilowatttime
<b>FHI</b>	Folkehelseinstituttet
<b>mmol/mol</b>	Millimol per mol
<b>mmol/L</b>	Millimol per liter
<b>CI</b>	Kontinuerlig integrasjon
<b>RMSE</b>	Root mean squared error



---

## Ordliste

<b>Glukose</b>	Glukose, eller blodsukker, måler innholdet av fritt sukker i blodet [1].
<b>Variasjon</b>	Variasjon i blodsukker beskriver hvor mye blodsukkeret til en diabetiker går opp og ned i løpet av en tidsperiode.
<b>Diskret variabel</b>	Diskret variabler er kvantitative variabler der kun enkelte tall langs tallskalaen er aktuelle kjennetegn [2].
<b>Standardnormalfordeling</b>	Standardnormalfordeling er en sannsynlighetsfordeling [2].
<b>Trend</b>	En trend indikerer at det er en langsiktig utvikling enten oppover eller nedover [2].
<b>Code review</b>	Code review omhandler at en eller flere personer, der minst en ikke har skrevet koden, skal lese og vurdere kildekoden til et program [3].
<b>Model-view-viewmodel</b>	MVVM er en programvarearkitektur som i større grad separerer det grafiske brukergrensesnittet og backend logikken, slik at koden ikke avhenger av en spesifikk modell platform [4].
<b>Bugs</b>	Bugs er en kodefeil i et program.
<b>Coupling</b>	Koblinger mellom forskjellige deler av koden, for eksempel mellom klasser. Ønsker så lav coupling som mulig [5].
<b>Cohesion</b>	Mengden oppgaver som en del av koden har ansvar for, for eksempel innad i en klasse. Ønsker høy cohesion, for eksempel at en klasse kun er ansvarlig for logikken som representerer den klassen [5].

---

# 1 Introduksjon og relevans

Denne seksjonen vil være en introduksjon til rapporten. Seksjonen vil presentere bakgrunnen for behov samt inspirasjonen for oppgaven. I tillegg vil formuleringen av forskningsspørsmålene bli presentert.

## 1.1 Bakgrunn for behov

Diabetes er en tidskrevende sykdom som kan ha alvorlige konsekvenser dersom den ikke blir tatt godt vare på. Dette kan for eksempel være ketoacidose, som betyr at blodet i kroppen blir livstruende surt [6]. Formålet med denne oppgaven er å simplifisere hverdagen til diabetikere ved å lage et rammeverk i form av en MVP for bokføring, og en algoritme som beregner korrekt insulinmengde og lærer av bokføringen. Målet er at algoritmen både skal beregne en mer korrekt insulinmengde, og at MVP-en skal være tidsbesparende for sluttbruker.

MVP-ens inspirasjon kommer fra testpersonens erfaring. Testpersonen har erfart at insulinbehovet varierer utover dagen, og at det ikke bare er karbohydratene i maten som har innvirkning på insulinmengden, men også hva slags type mat det er. Testpersonen hadde i månedene før testperioden manuelt justert hvor mye insulin de satte basert på tid på dagen. Dette i samme format som MVP-ens dagsprofiler 3.7.1. Basert på dataen lagret i testpersonens sensor applikasjon, LibreLink, kom det tydelig frem at dette medførte et bedre regulert blodsukker, men at det var mye ekstra arbeid med bokføringen av verdiene. Derav kom idéen om MVP-en som et bokføringsverktøy. Videre kom tanken om at en algoritme kunne styre disse verdiene, skalarene, slik at det ikke var behov for testpersonen å manuelt bokføre og analysere verdiene og blodsukkeret. Algoritmen ville dermed gjennomføre en analyse, slik at man hadde muligheten til å analysere både mer, og mer spesifikke aspekter i tillegg.

## 1.2 Problemstilling

Forskningsspørsmålet som blir tatt utgangspunkt i er:

- Hvor presist kan en algoritme basert på tidligere loggført data beregne en korrekt insulinmengde for å oppnå ønsket blodsukker verdi?
- Hvor mye tid kan sluttbruker spare i uken ved å ta i bruk MVP-en?

## 1.3 Rapportens struktur

### Seksjon 1: Introduksjon og relevans

Denne seksjonen vil introdusere bakgrunnen til oppgaven, samt presentere problemstillingen.

### Seksjon 2: Teori og relevant litteratur

Denne seksjonen vil introdusere teorien som er nødvendig for å tolke og forstå resultatene som er blitt funnet.

---

### **Seksjon 3: Metode**

Metodeseksjonen vil beskrive prosessen og utførelsen, samt valg av teknologi.

### **Seksjon 4: Resultater**

I denne seksjonen vil de vitenskapelige-, ingeniørfaglige og administrative resultatene bli presentert.

### **Seksjon 5: Diskusjon**

Denne seksjonen vil diskutere og drøfte resultatene fra oppgaven.

### **Seksjon 6: Konklusjon og videre arbeid**

I denne seksjonen presenteres de konklusjoner som kan trekkes i forhold til stilte problemstillinger, gitt diskusjonen i forrige seksjon, samt videre arbeid.

### **Seksjon 7: Samfunnspåvirkning**

Samfunnspåvirkning vil diskutere arbeidet i forhold til et helhetlig systemperspektiv. Resultatene blir sett på fra et miljømessig, helsemessig, samfunnsmessig og økonomisk perspektiv.

---

## 2 Teori og relevant litteratur

### 2.1 Diabetes type 1

Diabetes type 1 er en autoimmun sykdom. Dette vil si at kroppens immunforsvar ødelegger de produserende betacellene i bukspyttkjertellen [7] [8]. Bukspyttkjertellen har da en for lav eller ikke-eksisterende produksjon av insulin [9]. Grunnet dette er man selv nødt til å sette riktig insulinmengde. Å administrere diabetes type 1 er en omfattende prosess som involverer helsepersonell, og som krever kontinuerlig oppmerksomhet på flere områder. Dette er blant annet insulin administrasjon, overvåking av blodsukker, måltidsplanlegging, og prøvetaking for diabetes-relaterte konsekvenser [10]. Disse konsekvensene kan, som nevnt tidligere, være ketoacidose, eller hypoglykemi også kalt føling, som opptrer ved lavt blodsukker [11].

Dersom blodsukkeret synker under tre mmol/L vil kroppen varsle om at den behøver karbohydrater for å unngå at blodsukkeret synker mer. Symptomene som varsler er ofte hjertebank, svette, rask puls, uro, og angst [11]. Dersom man ikke får tilført karbohydrater og blodsukkeret fortsetter å synke, kan dette føre i verste fall føre til koma eller død, da det ikke er nok sukker igjen i blodet til å ernære hjernecellene [12]. Som regel vil et svært lavt blodsukker kun føre til at personen kommer i mental ubalanse og er sinnsforvirret frem til blodsukkeret stiger igjen med tilførte karbohydrater [12].

HbA1c er en type hemoglobin der glukose er bundet til hemoglobins betakjede [13]. HbA1c-verdien brukes for å undersøke kvaliteten av diabetes behandling [14] og oppfølging av denne sykdommen, og verdien gir informasjon den gjennomsnittlige blodglukosen de siste fire til tolv ukene [13]. Denne verdien måles i mmol/mol, men kan også oppgis i prosent. Personer med diabetes har en høyere HbA1c-verdi enn det som er normal. Den normale verdien for personer uten diabetes ligger vanligvis under 42 mmol/mol, som tilsvarer 6,0%, mens for diabetikere ligger verdien over 48 mmol/mol, 6,5%. HbA1c-verdien brukes for å undersøke hvor godt regulert en diabetiker sitt blodsukker er. Ettersom verdien er tett bundet opp til hvor godt eller dårlig et blodsukkeret er regulert, er den svært sentral i oppfølgingen av diabetes. Personer med diabetes får HbA1c-verdien målt ofte tre til fire ganger årlig, og det er normalt at denne verdien da er omtrent 53 mmol/mol eller 7,0%. 53 mmol/mol er også behandlingsmålet beskrevet av NHI og Noklus [15] [16]. Det kan være svært vanskelig å oppnå denne verdien, og i 2019 var det under 30% av diabetikere med diabetes type 1 i Norge som hadde en HbA1c-verdi under 53 mmol/mol [15]. Å ha en høy HbA1c-verdi gir en høyere risiko for diabetiske senkomplikasjoner. Dette er da konsekvensene av et høyt blodsukker på nervesystemet og blodårene [17].

Insulindosering er en konstant justering, der man ønsker å oppnå et lavt nok blodsukker til å minimere sjansen for komplikasjoner som ketoacidose, men samtidig unngå et farlig lavt blodsukkernivå. Normalt vil konsentrasjonen av blodsukker i blodet hos mennesker uten diabetes, være rundt fire - seks mmol/L [11], men det er også normalt mellom fire og åtte mmol/L [18]. Denne verdien kan øke dersom man spiser mat, og synke ned til tre til fire mmol/L ved lengre faste [11]. Hos diabetikere vil konsentrasjonen av blodsukker ofte være rundt 8,7 mmol/L [19]. Dette tallet er basert på sammenhengen mellom HbA1c-verdi og blodsukker, da den omtrentlige HbA1c-verdien hos diabetikere er cirka 53 mmol/mol gir dette et estimert gjennomsnittlig blodsukker på 8,7 mmol/L.

---

Insulinresistans på morgenen kalles «Dawnfenomenet» [20] [21]. Økningen i morgenblodsukkeret skyldes utskillelse av veksthormon, kortisol, glukagon og adrenalin i det kroppen forbereder seg på å våkne. Disse hormonene får leveren til å frigjøre glukose, og fenomenet forekommer vanligvis etter åtte til ti timers søvn [22]. Ettersom kroppen er mer insulinresistant på morgenen, har ofte diabetikere behov for en større mengde insulin enn ellers i løpet av dagen. For å beregne hvor mange enheter man behøver til frokostdosen, kan man ta i bruk 300-regelen. Denne regelen er til for å motvirke Dawnfenomenet. 300-regelen sier at man skal dividere 300 på den totale mengden insulin man bruker i løpet av et døgn. Tallet man får fra denne divisjonen sier hvor mange gram karbohydrater én enhet insulin tar hånd om [23].

Som diabetiker er man nødt til å undersøke hvilken sensitivitet man har til insulin. Dette gjelder da ulike typer scenerier som for eksempel korrigeringsinsulin og insulin til måltider. Korrigeringsinsulin er insulin som korrigerer blodsukkeret ned fra det som blir sett på som en for høy verdi. Dersom blodsukkeret til en diabetiker er over 8 mmol/L, kan man vurdere å sette korrigeringsinsulin [24]. Man kan ta i bruk 100-regelen for beregningen av korrigeringsinsulin. 100-regelen går ut på at man dividerer 100 på den totale døgndosen insulin. Dette tilsvarer da hvor mye en enhet insulin senker blodsukkeret [24].

For å beregne hvor mange insulin enheter som trengs til en viss mengde karbohydrater, kan man ta i bruk 500-regelen. Regelen sier at man dividerer 500 på den totale døgndosen med insulin, og dette gir antall gram karbohydrat én enhet insulin tar hånd om [24].

Det er flere ulike aspekter i hverdagen som kan påvirke insulinsensitivitet. Dette er blant annet stress, sykdom og trening [25] [26]. Ved langvarig stress kan man oppleve at man har et større behov for insulin og at det medfører et høyere blodsukkernivå. Ved akutt stress kan man derimot oppleve lavt blodsukker og hypoglykemi [25]. Dersom man er syk, enten med feber, influensa eller andre sykdommer, vil man også kunne oppleve et økt insulinbehov. Grunnen til at blodsukkeret stiger er at kroppen produserer stresshormoner som øker blodsukkeret [25]. Dersom man har diabetes type 1 kan blodsukkeret synke mens man trener, og grunnen til dette er at trening og fysisk aktivitet gjør at glukoseopptaket i kroppen øker som medfører at insulinet fungerer bedre. Det er derfor viktig for diabetikere å passe på at blodsukkeret er på en god blodsukkerverdi, gjerne seks til ti mmol/L, før trening, for å unngå lavt blodsukker eller hypoglykemi [26].

Variasjon i blodsukker beskriver hvor mye blodsukkeret til en diabetiker går opp og ned i løpet av en tidsperiode. Dersom man har stor variasjon, vil det være kraftige svingninger i blodsukkeret, og dette er ikke ideelt. Man kan for eksempel gå fra en svært høy verdi, til en svært lav verdi, kontinuerlig gjennom dagen. Det gjennomsnittlige blodsukkeret kan da være på rundt 8 mmol/L, noe som er en normal verdi for diabetikere, og det er derfor det er viktig å undersøke blodsukkerets variasjon. Å gå fra høye verdier til lave verdier kan være slitsomt for kroppen, i tillegg til at man ikke er godt regulert med slike svingninger. Grunnet dette er det derfor svært viktig å undersøke variasjonen i blodsukkeret. Variasjonen bør ligge under 36%. En godt anbefaling er å holde blodsukkeret mellom 3.9 og 10.0 mmol/L, minst 70% av døgnet [27].

---

### 2.1.1 Insulin

Insulin regulerer blodsukkeret i kroppen, og dersom man har diabetes type 1, er man nødt til å få tilført insulin, da kroppen enten ikke produserer dette, eller produserer for lite insulin [9]. Det finnes tre ulike hovedtyper insulin preparater, og disse er hurtigvirkende, middels langtidsvirkende, og langtidsvirkende insulin.

Hurtigvirkende insulin blir satt og tilpasset til hvert måltid. I tillegg brukes hurtigvirkende insulin for å korrigere et høyt blodsukker [28]. Effekten av dette varer i omtrent tre timer, basert på hva slags type insulin det er. For eksempel vil NovoRapid, et hurtigvirkende insulin, ha en maksimal effekt som varer i én til tre timer, men den totale virketiden er på rundt tre til fem timer [29]. Det finnes også hurtigvirkende insulin som varer til fire til seks timer [30].

Middels langtidsvirkende settes én til to ganger i løpet av dagen, og effekten av disse varer omtrent i 14 - 16 timer [30]. Langtidsvirkende insulin blir satt én gang om dagen, og effekten av dette varer i minst 24 timer. Uten insulin i kroppen, vil blodsukkeret øke. Middels langtidsvirkende og langtidsvirkende insulin brukes for å holde blodsukkeret jevnt og regulert gjennom dagen, natten og mellom måltider [28].

## 2.2 Statistikk

### 2.2.1 Enkel lineær regresjon

Regresjonsanalyse går ut på å finne ut hvilken sammenheng det eksisterer mellom variablene. Enkel lineær regresjon går ut på å se på kun en rettlinjet sammenheng mellom to variabler [2].

### 2.2.2 Regresjonsmodellen

En rett linje kan defineres som:

$$y = \alpha + \beta x \quad (1)$$

der  $\alpha$  er punktet linjen krysser y-aksen, og  $\beta$  er linjens stigningstall. Regresjonsanalysens hensikt er å finne det beste estimatet av den ukjente linjen  $y = \alpha + \beta x$  som beskriver den modellmessige sammenhengen mellom forklaringsvariabelen og responsvariabelen. Det beste gjettet på den ukjente linjen er estimatet av de ukjente størrelsene  $\alpha$  og  $\beta$  på bakgrunn av de innsamlede dataene, og kalles regresjonslinjen [2]:

$$\hat{y} = \hat{\alpha} + \hat{\beta}x \quad (2)$$

### 2.2.3 Regresjonslinjen og minste kvadratersmetode

Man ønsker å estimere regresjonslinjens koeffisienter  $\hat{\alpha}$  og  $\hat{\beta}$ . Denne estimeringen må basere seg på den innsamlede dataen. Regresjonslinjen  $\hat{y} = \hat{\alpha} + \hat{\beta}x$  er linjen som passer datasettet best. Løsningen man ofte bruker for å finne regresjonslinjen, kalles *minste kvadraters metode*. Minste kvadraters metode tar utgangspunkt i en vilkårlig linje. For hver punkt kvadrerer man avstanden mellom punktet og linjen, og man finner *avvikskvadratet*. Deretter finner man summen  $K$  av arealene til alle

---

avvikskvadratene. I minste kvadraters metode skal man så velge den linjen som gir en minst mulig kvadratsum  $K$  [2].

Dataene  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  foreligger, og man skal tilpasse linjen  $\hat{y} = \hat{\alpha} + \hat{\beta}x$  som passer best til observasjonsparene. Minste kvadraters metode vil gi de følgende koeffisientene  $\hat{\beta}$  og  $\hat{\alpha}$ :

$$\hat{\beta} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (3)$$

$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x} \quad (4)$$

#### 2.2.4 Varians

Varians er den gjennomsnittlige størrelsen på avvikskvadratene. Variansen til Y-ene blir definert som:

$$S_Y^2 = \frac{1}{n-1} \sum_{i=1}^n (Y_i - \bar{Y})^2 \quad (5)$$

der  $n$  er antall observasjoner [2].

Den beste gjetningen man har på variansen  $\sigma^2$  er lik:

$$s^2 = \frac{SS_E}{n-2} \quad (6)$$

$SS_E$  blir beskrevet i delseksjon 2.2.6.

#### 2.2.5 Standardavvik for lineær regresjon

Et typisk avvik fra gjennomsnittsverdien kalles standardavvik. Standardavvikets utgangspunkt er å se hvor mye hver enkelt observasjon avviker i forhold til gjennomsnittet [2]. Standardavvik er kantlengden til det gjennomsnittlige avvikskvadratet. Standardavviket defineres som kvadratrotten av variansen:

$$s = \sqrt{\frac{SS_E}{n-2}} \quad (7)$$

der  $n$  er dataparene samlet inn.

#### 2.2.6 Regresjonsmodellens godhet

Summen av kvadratene i ligning 5 mellom Y-observasjonene og deres gjennomsnittsverdi, kalles  $SS_T$ . Dette står for det engelske *total sum of squares*. Fra dette kan man definere ytterligere to slike summer av avvikskvadrater.  $SS_R$  er summen av kvadratene mellom regresjonslinjen og gjennomsnittet av Y-ene [2].  $SS_E$  er den minste verdien til kvadratsummen  $K$ , der  $K$  er summen av alle kvadratene rundt regresjonslinjen. Dette gir følgende sammenheng:

$$SS_T = SS_R + SS_E \quad (8)$$

---

som da er:

$$\sum_{i=1}^n (Y_i - \bar{Y})^2 = \sum_{i=1}^n (\hat{\alpha} + \hat{\beta}x_i - \bar{Y})^2 + \sum_{i=1}^n (Y_i - \hat{\alpha} - \hat{\beta}x_i)^2 \quad (9)$$

Den totale variasjonen blant Y-ene ( $SS_T$ ) kan deles opp i i to deler, der del én forklares av modellen ( $SS_R$ ) og den andre delen som kun skyldes tilfeldige avvik ( $SS_E$ ). Dersom modellen forklarer store deler av variasjonen, kan man anta at modellen er god. Andelen av variasjonen som forklares av modellen er:

$$r^2 = \frac{SS_R}{SS_T} \quad (10)$$

### 2.2.7 Forventning

Dersom man observerer verdien av en stokastisk variabel  $X$  i en stor rekke stokastiske forsøk, vil man kunne regne gjennomsnittet for datasettet som forsøkene vil resultere i. Man kan derimot regne ut et teoretisk gjennomsnitt med utgangspunkt i sannsynlighetsfordelingen. Dette beregnede "gjennomsnittet" for  $X$  er variabelens forventningsverdi [2]. Forventningsverdien til en diskret variabel  $X$  defineres som:

$$\mu = E(X) = \sum_{\text{alle } x_i} x_i * P(X = x_i) \quad (11)$$

### 2.2.8 Konfidensintervall for ukjent linje

Området langs tallinjen som man, ut fra statistisk teori, vet har en høy sannsynlighet for å inneholde en tallfestet egenskap ved populasjonen kalles et konfidensintervall [2].

Det finnes ulike typer konfidensintervall: Z-intervall og T-intervall. Man benytter seg av Z-intervall når både standardavviket  $\sigma$  og forventning  $\mu$ . T-intervall brukes når forventning  $\mu$  er kjent og standardavviket  $\sigma$  er ukjent. I tilfellet er standardavviket  $\sigma$  er ukjent, kan man ikke ta utgangspunkt i den standardnormalfordelte Z. Man må ta utgangspunkt i T-variabelen:

$$T = \frac{\bar{X} - \mu}{\frac{s}{\sqrt{n}}} \quad (12)$$

### 2.2.9 Prediksjonsintervall for observasjonene

Prediksjonsintervall, også kalt spredningsintervall, beskriver hvilke observasjoner som er sannsynlige. En enkelt observasjon er mer usikker enn gjennomsnittet av alle observasjonene. Denne usikkerheten er knyttet til hvor den ukjente linjen skal være. I tillegg må man ta hensyn til usikkerheten som kommer av feilledet  $e$ . Av denne grunnen kan man anta at prediksjonsintervallet er bredere enn den ukjente linjens intervall. Et 95% preiksjonsintervall for enkeltobservasjonen  $Y$  for en gitt  $x$ -verdi er lik:

$$\hat{\alpha} + \hat{\beta}x \pm t_{0.025} * s \sqrt{1 + \frac{1}{n} + \left(\frac{x - \bar{x}}{s \cdot SE(\beta)}\right)^2} \quad (13)$$



---

Dersom prediksjonsintervallet er svært bredt, er det liten grunn til å lage prediksjoner. Et bredt prediksjonsintervall henger sammen med en lav  $r^2$ -verdi. Dette betyr at regresjonslinjen forklarer kun en liten del av variasjonen i datamaterialet. I tillegg henger bredden av prediksjonsintervallet sammen med antall observasjoner [2].

### 2.2.10 Kovarians og korrelasjon

For å undersøke om det eksisterer en sammenheng mellom to variable,  $X$  og  $Y$ , der det ikke er tilfeldig hvordan de varierer i forhold til hverandre kan man bruke kovarians [2].

$$Cov(X, Y) = E(X * Y) - \mu_x * \mu_y E(X * Y) = \sum_{all x_i} \sum_{all y_i} x_i * y_i * P(x_i, y_i) \quad (14)$$

Dette gir en tallverdi, som ofte kan være vanskelig å tolke. Grunnet dette «normaliserer» man kovariansen ved å dele på variabelens standardavvik. Dette blir kalt for korrelasjonen mellom variablene [2]. Formelen for korrelasjon er:

$$\rho(X, Y) = \frac{Cov(X, Y)}{\sigma_x * \sigma_y} \quad (15)$$

Korrelasjonen vil alltid ligge mellom -1 og 1, da den er normaliser. Absoluttverdien til korrelasjonen,  $\rho$ , indikerer hvor sterk den lineære sammenhengen mellom de to variablene er. Dersom absoluttverdien er høy, vil dette bety en større korrelasjon. I tillegg avgjør fortegnet retningen på sammenhengen. Dersom det eksisterer en positiv korrelasjon, betyr dette at dersom den ene øker vil den andre også øke. Hvis det derimot er en negativ korrelasjon betyr dette at dersom den ene øker vil den andre parten synke [2].

### 2.2.11 Interrupted time series

Statistisk tidsrekkeanalyse er en teknikk som blir brukt for å analysere hvordan en variabel  $Y$  utvikler seg over tid. Målingene blir ofte tatt på like intervaller over tid [2]. ITS analyse er et svært godt design for å evaluere effektiviteten av helseintervensjoner på befolkningsnivå som har blitt implementert på et klart definert tidspunkt [31]. I en slik analyse brukes en tidsrekke av et bestemt utfall av interesse for å etablere en underliggende trend. Denne trenden blir "avbrutt" av en intervensjon på et kjent tidspunkt. Denne intervensjonen kan modelleres på flere måter, basert på hvordan intervensjonen er forventet å påvirke dataene. Den generelle formelen ser man under [31].

$$Y_t = \beta_0 + \beta_1 T + \beta_2 X_t + \beta_3 T X_t \quad (16)$$

$\beta_0$ - og  $\beta_1$ -leddene utgjør basen og er ekvivalent med den underliggende linjen, hvor  $T$  er tiden.  $\beta_2$ -leddet er et nivåskifte, da  $X_t$ -variabelen bare er en dummy-variabel, 0 før intervensjonen og 1 etter.  $\beta_3$ -leddet vil også først bli introdusert etter intervensjonen, men dette vokser med tiden  $T$ . For å modellere en momentan endring kan man fjerne  $\beta_3$ -leddet, og man sitter igjen med en linje som har et nivåskifte ved intervensjonen.

---

### 2.2.12 Root mean squared error

Når man har et sett med punkter (x- og y-koordinater) og man ønsker å tilpasse en modell for dette settet, for eksempel med ITS 2.2.11. Feilen mellom den oppnådde modellen og hvert punkt er i en vektor  $r$ . For å finne RMSE verdien til modellen bruker man følgende formell:  $RMSE = \frac{\sqrt{r_1^2 + \dots + r_m^2}}{\sqrt{m}}$  [32]. Om  $r$  vektoren inneholder bare nuller vil det si at feilen til hvert punkt er null, dette gir RMSE lik null, da en perfekt modell. RMSE-verdiene til ulike modeller kan sammenlignes for å se hvilken modell som beskriver settet med punkter best, da den med RMSE-verdi nærmest null.

## 2.3 Algoritme konstruksjon

Algoritme konstruksjon og design blir beskrevet som en prosess der man kommer opp med en ide for en beregningsmessig gjennomførbar teknikk for å oppnå en ønsket adferd [33]. Denne prosessen består av problemløsning, kunnskap om algoritme konstruksjon og kunnskap innenfor det spesifikke emnet, i dette tilfellet diabetes type 1. Følgene seks steg, hentet fra forskningsartikkelen «*Understanding and Automating Algorithm Design*» [33], blir ofte bruk i algoritme konstruksjon:

1. Forstå problemet
2. Planlegg en løsning rundt kjernen av idéen, og raffinér eller utdyp kjernestrukturen
3. Utfør den delvis spesifiserte algoritmen
4. Legg merke til og formuler eventuelle problemer eller muligheter
5. Bekreft at strukturen er en løsning
6. Evaluer løsningen

Det første steget omhandler å ha en god forståelse av problemet, og dette kan gjøres på ulike måter. For eksempel kan man bruke figurer for å visualisere utfordringen, eller ved å liste opp egenskapene til problemet [33].

Ved steg to starter designet av algoritmen rundt en kjerneide. Denne idéen kan variere basert på designerens erfaringer og kunnskap. For eksempel kan man bruke en strategi der man genererer og tester, eller bruke strategier som dynamisk programmering. Etter å ha utarbeidet en plan vil designeren raffinere stegene av kjerneideen. Her kan designeren dekomponere problemet i mindre komponenter, introdusere nye input og deklare detaljer rundt strukturen [33]. Stegene for finpussing kan foreslås av kunnskap innenfor feltet, eller kan være resultatet av å forsøke å utføre algoritmen. Dersom kunnskap ikke strekker til, kan man forsøke en søk og feil metode, der man tester/hypotiserer ulike algoritmesteg og tester de ved å utføre den delvis spesifiserte algoritmen.

Ved å utføre den delvis spesifiserte algoritmen, steg tre, kan man trekke konklusjoner rundt algoritmen som er blitt utviklet så langt. Når algoritmen testes er det lettere å finne eventuelle problemer eller muligheter ved den. I dette steget kan man også utføre algoritmen ved å endre på hvilke algoritmesteg som blir utført først for å finne eventuelle styrker og svakheter [33].

---

Som følge av å utføre den delvis spesifiserte algoritmen kan ulike utfordringer og problemer oppstå. Disse kan bli synlagt gjennom utførelsen ved at man oppdager at man mangler noen steg i algoritmen, eller at man oppdager at deler av algoritmen ikke stemmer overens. Det er også mulig at prosessen kan avdekke muligheter for optimalisering eller forbedringer. Dette kan for eksempel være at man oppdager forbedringer i en tidlig skisse som man ikke så etter da man først lagde visualiseringen. Man har større sannsynlighet for å oppdage slike muligheter dersom man har gode kunnskaper innenfor fagfeltet algoritmen utvikles [33].

Steg fem kan gjennomføres ved å teste algoritmen på enkle og spesifikke verdier og observere om noe uventet oppstår. I dette steget vil man også måtte ta hensyn til enkelttilfeller [33].

I det siste steget vil man evaluere løsningen, og dette kan gjøres på ulike måter. Et eksempel er å sammenligne den med et allerede eksisterende algoritme alternativ. Man kan også utføre en kompleksitetsanalyse av algoritmen [33].

## 2.4 Systemutviklingsmetodikk

Systemutvikling er en måte å arbeide målrettet og metodisk på, ofte for å utarbeide nye datasystemer. Systemutviklingen inneholder flere ulike aktiviteter som for eksempel utvikling, testing, og kravarbeid. En systemutviklingsmetode er en måte å organisere disse aktivitetene på. Det finnes flere ulike systemutviklingsmetoder, og blant disse finner man blant annet vannfallsmetoden. Denne metoden var den vanligste i flere år, og la stor vekt på blant annet dokumentasjon, og alle prosessene ble utført sekvensielt [34]. Dette er ofte en lang og tidskrevende metodikk, samt at endringer blir svært dyre dersom kravene til systemet endres underveis. I 2001 kom *The Agile Manifesto* [35], og dette ble utgangspunktet for smidig eller agil systemutviklingsmetodikk. Hovedprinsippene i manifestet er:

- Personer og samspill fremfor prosesser og verktøy
- Programvare som virker fremfor omfattende dokumentasjon
- Samarbeid med kunden fremfor kontraktsforhandlinger
- Å reagere på endringer fremfor å følge en plan

*The agile manifesto* legger mer vekt på god kommunikasjon, fleksibilitet og hyppige leveranser, og man står dermed i en bedre posisjon til å endre kravene underveis, i motsetning til vannfallsmetoden. Noen populære smidige systemutviklingsmetoder er Scrum og Kanban.

## 2.5 Brukertester

Den typiske brukertesten er en test der man inviterer en bruker til å teste en prototype eller en applikasjon man har utviklet. Dette er for å undersøke hvor intuitivt og godt designet er, og finne diverse problematiske design valg eller feil. Man kan bruke brukertestene til å evaluere designet, og forbedre det basert på testene. Normalt vil man brukerteste et design flere ganger, gjøre endringer, og deretter gjennomføre nye brukertester av det nye designet. Dette er for å bekrefte og sikre at det nye designet

---

har forbedret brukerens opplevelse, og ikke introduserte nye feil. Man gjennomfører ofte disse testene på personer i målgruppen til prototypen eller applikasjonen [36].

## 2.6 Sensor, LibreLink

FreeStyle Libre sensoren er en sensor teknologi som gir kontinuerlig oppdaterte glukoseverdier, samt at disse blir loggført. Disse kan hentes ut gjennom LibreLink-applikasjonen på en mobil enhet. Det tilbys også grafiske visualiseringer og statistikk om tiden innenfor et gitt glukoseintervall, gjennomsnitts glukose, estimat til HbA1c-verdi og lignende.

## 2.7 Tidligere relevant arbeid

Denne seksjonen vil presentere tidligere relevant arbeid innenfor fagfeltene som denne oppgaven har undersøkt og brukt, blant annet statistikk og algoritmer for diabetes.

ITS har gjentatte ganger blitt brukt for å evaluere intervensjoner for befolkningens helse. Blant disse finner man blant annet forskningsartiklen «*Interrupted time series regression for the evaluation of public health interventions: a tutorial*» [31], og artikkelen som diskuterer bruken av ITS i helsestuder «*Use of Interrupted Time Series Analysis in Evaluating Health Care Quality Improvements*» [37].

Det er blitt utarbeidet lignende teknologi, ment for å hjelpe leger og sykepleiere involvert i insulin dosering av pasienter på sykehus, kalt GlucoTab. GlucoTab er en algoritme som automatisk estimerer forslag til insulinmengder for pasienter med diabetes type 2. GlucoTab tillater digital planlegging, implementasjon og dokumentasjon for alle nødvendige aspekter for å regulere blodsukkeret. Det er en mobilapplikasjon, og systemet er utviklet av ingeniører, leger og sykepleiere. Systemet er ment som et verktøy for å hjelpe helsepersonell med dosering av insulin for innlagte pasienter [38].

I tillegg til at det har blitt undersøkt algoritmen for diabetes type 2, er det også algoritmer for diabetes type 1. Det eksisterer blant annet en iterativ algoritme som bruker gårsdagens glukose, insulin og måltidsdata, for å beregne insulin. Denne algoritmen er beskrevet i mer detalj i forskningsartikkelen «*A Model-Based Insulin Dose Optimization Algorithm for People With Type 1 Diabetes on Multiple Daily Injections Therapy*» [39]. Resultatene fra dette artikkelen var svært lovende, og forskningsartikkelen har vist at det er mulig å regulere blodsukkeret med en iterativ algoritme.

---

## 3 Metode

### 3.1 Forskningsmetode

Forskningsmetode er ulike fremgangsmåter som blir benyttet i vitenskaplig forskning [40]. Grunnen til at forskningsmetode er viktig, er fordi det brukes for å etablere pålitelig kunnskap. Forskningsmetoden som ble tatt i bruk i dette prosjektet startet med utgangspunkt i egne erfaringer med diabetes, og motivasjon for å gjøre denne sykdommen lettere å håndtere. Deretter ble det formulert to forskningsspørsmål. Etter dette ble det brukt ulike strategier for å svare på forskningsspørsmålet. Disse besto av design og kreasjon av MVP-en, eksperimenter med tidligere eksisterende glukosedata for å undersøke hva slags type statistiske modeller som kunne passet, og en testgruppe bestående av det ene medlemmet i gruppen. Denne testpersonen ble det utført en case study på. Deretter ble evaluering av «design og kreasjon» og «case study» brukt for å generere den nødvendige dataen for algoritmen og analysen. Observasjon ble brukt for å generere data angående brukergrensesnittet, i tillegg til at det ble gjennomført to brukertester. Brukertestene går da under data genereringsmetoden «Intervju». For å analysere dataen, ble det tatt i bruk både kvalitative og kvantitative metoder. Analysen er kvalitativ basert på brukeropplevelsen av MVP-en, og kvantitativ på dataen samlet inn fra MVP-en gjennom analysen. Forskningsmetoden som ble tatt i bruk under dette prosjektet blir vist i vedlegg A.

Fordeler ved egne erfaringer og motivasjon, var at det ga gruppen et klart mål om hva de ønsket å oppnå og hvordan dette skulle gjøres. Forskningsspørsmålene som ble formulert hjalp gruppen med å spesifisere fokuset i oppgaven, og dette hjalp gruppen å prioritere arbeidsoppgaver og planlegge prosessen. En ulempe gruppen var klar over var at dersom forskningsspørsmålene ble dårlig formulert, eller vage, ville det være vanskelig å konkludere. Ved å eksperimentere med eksisterende glukosedata, ga dette gruppen et godt innblikk i hva slags type data de trengte for å besvare problemstillingen. Strategien «design og kreasjon», ga gruppen mulighet til å tilpasse systemet slik at de fikk gjennomført case studyen som ønsket. I tillegg kunne gruppen legge til rette for hvordan de ville samle inn data. Ulempen ved denne strategien er at prosessen tar svært lang tid, og vil utsette datagenereringen og -innsamlingen. Fordelen med strategien «case study», er at denne gir det mest naturlige testmiljøet, og de beste tilbakemeldingene både på algoritmen og MVP-en sitt design. En case study tar derimot mye tid, og kan by på ulike problemer, for eksempel dersom det skulle oppstå bugs i koden. Metoden for å generere data angående brukergrensesnittet og designet gjennom intervju og observasjon, har en fordel ved at man får samlet inn data relativt raskt, gjennom for eksempel brukertester. Dette er da data man kan bruke for å forbedre designet. Ulempen ved intervju som brukertester, er at disse ofte må gjentas, dersom det skjer en endring i designet, for å forsikre at designet ikke ble dårligere. Evaluering av «design» og «case study» ble brukt for å generere data, ved at testpersonen fikk utprøvd den designede MVP-en og algoritmen. Ulempen ved dette er at det kan ta lang tid, da testingen sin varighet er svært lang. Metoden ble imidlertid valgt, da det var essensielt å samle inn den nødvendige dataen for algoritmen. For å analysere den innsamlede dataen på brukergrensesnitt ble det tatt i bruk kvalitative analyse. Fordelen med slike tester er at man, for eksempel, får undersøkt om designet er intuitivt. Ulempen er at disse testene kan være subjektive, og det er derfor viktig å gjennomføre nok tester for å få et helhetlig innblikk i resultatene. Ved den kvantitative analysen får gruppen svar på forskningsspørsmålene. Det er viktig å

---

skaffe disse svarene på en objektiv og metodisk måte. Det var derfor helt nødvendig med en kvantitativ analyse gitt forskningsspørsmålene.

## 3.2 Prosess

I starten av prosjektet ble det satt opp seks milepæler. Disse var:

- Betaversjon/førsteutkast
- Lage poster
- Implementere algoritmen for insulinmengde
- Lage og teste resultatanalyse
- Gjennomføre resultatanalyse
- Hovedrapport

Disse milepælene representerer det gruppen anså som viktige hendelser under prosjektet. Milepælen «Lage og teste resultatanalyse» ble imidlertid fjernet av gruppen, da de ikke følte dette var en naturlig milepæl lenger ut i prosjektet. Gruppen brukte i stor grad Gantt-diagrammet for oversikt over progresjonen til prosjektet, og milepælene ble brukt mer som symbolske punkter. Et eksempel på dette er «Lage poster», som opptrådte rundt halvveis i prosessen, og fungerte godt for å representere prosjektets daværende progresjon, og hva som gjensto av arbeid.

### 3.2.1 Informasjonsinnhentning

Ved prosjektets start ble det besluttet at gruppemedlemmene skulle gjennomføre et kurs i Xamarin Forms. Ettersom det ene gruppemedlemmet allerede hadde gjennomført dette kurset, gikk dette medlemmet kun gjennom de delene av kurset som trengte oppfriskning. Grunnen til at gruppen valgte å gjennomføre dette kurset var for at begge medlemmene skulle få en lik teoretisk og praktisk innsikt i teknologien som skulle tas i bruk. Et annet alternativ til å ta kurset var at begge medlemmene leste seg opp på dokumentasjonen til Xamarin Forms, men dette ble sett på som tungvindt og mindre effektivt enn kurset. I tillegg kunne dette ført til at gruppemedlemmene hadde ulik kunnskap og forståelse for teknologien, som kunne ført til konflikter under utviklingen. Gruppen konkluderte med at selv om kurset tok mye tid, var dette tid som ville bli spart inn ved å ikke måtte lete rundt i og lese dokumentasjonen for de nødvendige elementene for MVP-en. Det ble ansett som det beste alternativet å ta kurset, da det ene gruppemedlemmet allerede var klar over hva slags informasjon det formidlet, og etter diskusjon innad i gruppen, ble det bestemt at det var denne informasjonen og kunnskapen som trengtes for å utvikle MVP-en.

### 3.2.2 Testgruppe

I startfasen av prosjektet ble det etablert at det var behov for en testgruppe, men at dette kunne by på utfordringer rundt tidsbruk. Det ble avklart med veileder at det var nok med én testperson for en MVP. Deretter fra den preliminnære dataen kan man

---

konkludere om det er verdt å videreutvikle MVP-en til en fullverdig applikasjon, og bruke en større testgruppe for å se om flere får lignende resultater. Testpersonen er Ola Kristoffer Hoff, en diabetiker, men ellers en frisk voksen mann på 21 år, som er en del av gruppens medlemmer. Fordelen med å kun ha en testperson var at det var færre personer å forholde seg til, samt at det bydde på færre utfordringer rundt distribusjonen av MVP-en. En ulempe er at gruppen ikke fikk undersøkt om algoritmen ville fungert for flere diabetikere, og at det ville føre til mindre preliminaire data. Likevel konkluderte gruppen med at det var nok med én testperson etter råd fra veileder.

### 3.2.3 Datainnsamling

Gruppen rådførte seg med «Statistikkhjelpen for bacheloroppgave i ingeniørfag» for å unngå fallgruver rundt datainnsamling. Dette var for å sikre dataens integritet. Disse fallgruvene besto av å fjerne ekstremalpunkter fra dataen i prosenter, størrelsen på datamengden, og lignende. Gruppen har ellers funnet ut av og laget alt av statistiske analyser og modeller selv.

Algoritmen trenger en nødvendig mengde data for å kunne gjøre riktige endringer for å kalkulere riktig insulinmengde. Tidlig i prosessen ble det identifisert et mulig problem rundt evnen til å samle inn nok data. Det ble vurdert å bruke allerede eksisterende datasett, men det kom fort frem at disse ikke var spesifikke nok, og at gruppen måtte samle inn den nødvendige dataen selv. For å samle inn denne dataen, måtte gruppen utvikle et rammeverk i form av en MVP til en applikasjon. Deretter tok testpersonen i bruk MVP-en på sin mobile enhet. Dataene fra MVP-en ble lagret i en database som deretter ble formatert og eksportert til CSV filer for analyse.

Som nevnt tidligere var gruppen svært avhengig av å anskaffe den nødvendige mengden data. Grunnet dette valgte gruppen å jobbe flere ekstra timer i starten av prosjektet for å få MVP-en for datainnsamlingen på plass så tidlig som mulig. Dette var for å forsikre at det ble anskaffet nok data, i tillegg til at gruppen hadde noe slingsmonn dersom det skulle oppstå uventede hendelser som for eksempel sykdom. Ettersom det skjedde endringer i prioriteringslisten over funksjonalitet og «events» ikke lenger var planlagt å bli implementert, se vedlegg D (prosessdokumentasjon), har ikke MVP-en tatt hensyn til hendelser som sykdom. Som nevnt i delseksjon 2.1 vil sykdom ha stor innvirkning på insulinsensitivitet. Grunnet dette så gruppen på data da testpersonen var syk, som korrupt, da disse datapunktene ville vært ekstremalverdier og påvirket algoritmen på en negativ måte så lenge slike hendelser ikke er tatt hensyn til.

Som nevnt i delseksjon 2.1.1 har hurtigvirkendeinsulin, basert på type, ulik virketid. Virketiden gruppen har tatt utgangspunkt i henhold til funksjonaliteten «påminnelser», er tre timer. Grunnen til at tre timer ble valgt er basert på artikler om insulinetsvirketid, om det spesifikke insulinet testpersonen bruker, NovoRapid, og dets virketid, samt erfaringer testpersonen har hatt med virketiden av hurtigvirkende insulin tidligere. I MVP-en vil det opprettes en påminnelse etter at testpersonen har registrert satt insulin. Denne påminnelsen blir satt til tre timer etter registreringen, grunnet insulinetsvirketid. Påminnelsen ber testpersonen om å skrive inn hvilken verdi blodsukkeret hadde tre timer etter siste loggførte måltid. Denne verdien blir lagret som det fastende blodsukkeret etter måltidet og viser hvor langt unna den ønskede verdien man oppnådde å regulere blodsukkeret. Det er denne differansen algoritmen

---

baserer seg på og gjør endringer ut ifra.

Et problem som ble identifisert tidlig i prosessen var overlappende måltider. Som nevnt i delseksjon 2.1.1 er insulin sin maksimale effekt i én til tre timer etter at dosen har blitt satt, og måltider som blir konsumert innenfor dette tidsintervallet blir sett på som overlappende måltider. Det er altså svært aktivt insulin i kroppen når man må sette mer insulin for å korrigere for maten som skal bli konsumert til det overlappende måltidet. En antagelse gruppen tar er at algoritmen beregner en korrekt mengde insulin, både til maten og korrigerer ved det første måltidet. Man antar altså at testpersonen ville ha endt opp på ønsket verdi etter tre timer. Dersom testpersonen har et overlappende måltid, vil man ta utgangspunkt i at man oppnådde ønsket verdi ved første måltid, og ta utgangs punkt i den ønskede verdiene i utregningen av insulin estimatet for det andre måltidet.

En svakhet ved denne metoden er at om første måltid hadde en stor feil så vil denne feilen også falle på det andre måltidet, som kanskje ikke hadde en feil. Dette anså gruppen som en liten fallgrube da kombinasjonen av de to (eller flere) måltiden sjeldent vil være helt like, hvert fall ikke ofte nok til at det får en stor effekt på det andre måltidet. Gruppen vurderte alternativer til å løse dette problemet, men gruppen kom frem til at dette ville være den beste løsningen, med bakgrunn i hvor begrenset mengden data var, og dette er metoden testpersonen har brukt manuelt tidligere.

### **3.2.4 Databehandling**

Det var nødvendig å behandle dataen slik at den kunne presenteres i grafer som senere kunne tolkes for resultatanalysen. Som nevnt i delseksjon 3.2.3, blir testpersonen sine data eksportert til CSV filer. Det var viktig tidlig i prosessen å sikre at dataene som ble lagret kunne anvendes i en analyse for å kvantitativt kunne vurdere algoritmen og MVP-ens innvirkning på blodsukkerreguleringen til testpersonen. Testpersonen bruker en sensor 2.6 som loggfører blodsukkerreguleringen på en god og oversiktlig måte, og testpersonen hadde mange måneder med data før test perioden som gruppen kunne bruke som referanse til verdiene loggført under testperioden. Dette ga et godt innblikk i blodsukkerreguleringen helt upåvirket av algoritmen og MVP-en. Videre så ble dataene fra selve MVP-en også brukt for å se om det var statistisk grunnlag for å si om algoritmen hadde hatt påvirkning på nøyaktigheten til insulinestimatet per matvare og dagsprofil. Dette kunne undersøkes for alle endringer gjort av algoritmen.

### **3.2.5 Sikkerhetsmekanismer**

Ettersom diabetes kan ha alvorlige konsekvenser, slik som beskrevet i delseksjon 2.1, var det viktig for gruppen å legge inn sikkerhetsmekanismer slik at algoritmen ikke hadde mulighet til å foreslå for store endringer av gangen. Dette grunnet at for store endringer kunne ført til alvorlige feildoseringer. Når algoritmen utfører analysen av dataene vil den foreslå en endring i dosering for å oppnå ønsket verdi, men det er lagt inn en streng grense. Denne grensen kalles «nedre grense for prediksjonsintervall», som sikrer at blodsukkeret ikke havner under ett mmol/L i glukoseverdi under ønsket oppnådd verdi, med en 95% sikkerhet. Dette gjøres ved et 95% prediksjonsintervall hvor den nedre linjen ikke kan havne under den strenge grensen på minus en. En



---

ulempe med dette er at algoritmen kan bli begrenset. Som kan føre til at blodsukkeret blir regulert til en høyere verdi enn ønsket for å holde seg over denne linjen, men gruppen så på det som veldig viktig å ha en konservativ og streng grense grunnet alternativet hvor doseringene ble for store for ofte, se delseksjon 2.1.

Det eksisterer også en annen sikkerhetsmekanisme som begrenser endringen algoritmen kan gjøre, «absolutt største justeringsfaktor». Denne sikkerhetsmekanismen har høyest prioritet, og vil overkjøre «nedre grense for prediksjonsintervall». Dette grunnet at et dårlig datasett kan gi et stort prediksjonsintervall som kan ønske å justere dataene drastisk. Denne sikkerhetsmekanismen vil begrense hvor stor denne endringen blir, med en maksimal endring på én enhet i den gitte analysen. Disse sikkerhetsmekanismene sikrer at algoritmen vil justere verdier gradevis, men også at bruker ikke skal være i fare for at algoritmen foreslår farlige doser. Denne begrensningen har en stor ulempe i at det vil ta flere iterasjoner enn det som kanskje var nødvendig for å justere verdien til det ønskede nivået, det vil også i tilfeller hvor den teoretiske verdien er langt unna den ønskede verdien ta lenger tid før den får korrigert seg, som kan medføre at den mavaren eller dagsprofilen kan skape dårlige verdier for brukeren.

### 3.2.6 Testing

For å sikre at koden fungerer som den skal valgte gruppen å ha testdekking av backend kode. Det ble vurdert å skrive tester for frontend kode, men for å gjennomføre dette måtte gruppen skrevet om all frontend koden, og endret arkitekturen til MVVM. Dette er en svært tungvind og tidskrevende prosess, erfart av gruppen i løpet av kurset. Gruppen konkluderte med at ettersom frontend for det meste kun var designet av MVP-en, og ikke noe kritisk logikk, ville det vært dårlig bruk av ressurser og tid å endre koden til denne arkitekturen. Det ble også ansett som viktigst å teste businesslogikken og all kode vedrørende algoritmen. Dette for å sikre at algoritmen fungerer som den skal. Gruppen ble derfor enig med veileder at det var kun nødvendig å skrive tester til business logikken og algoritmen. Den totale testdekningsprosenten var 47,38%.

Gruppen konkluderte også at det ikke var behov for kontinuerlig integrasjon, CI. Dette var fordi gruppen tok i bruk parprogrammering som sikret kontinuerlig code-review. I tillegg skrev gruppen testene etter at MVP-en var ferdigutviklet. Dette førte til at testene ble kjørt etterhvert som de ble skrevet. Gruppen vurderte å ha CI ved begynnelsen av prosjektet, men grunnet den kontinuerlige parprogrammeringen konkluderte medlemmene med at det var unødvendig.

### 3.2.7 Presisjon

Presisjonen til algoritmen blir basert på HbA1c-verdi over de fire siste ukene fra testperiodens slutt. I følge Noklus og NHI, er 53 mmol/mol, eller 7 prosent, det ønskede behandlingsmålet, og av den grunn har også gruppen valgt å markere dette som ønsket presisjon. HbA1c-verdien er altså algoritmens presisjon. Som nevnt i delseksjon 2.1, kan dette målet være svært vanskelig å nå for diabetikere med diabetes type 1.

---

### 3.2.8 Tidsbruk

Et viktig mål for prosjektet er å redusere tiden en diabetiker bruker på å regulere blodsukkeret. Da gruppen utarbeidet oppgaveteksten, ble det spesifisert at man skulle opprette matvarer og lagre informasjon om disse i en database. Brukeren skal få opp en liste med disse matvarene, og trykke på de den ønsker å spise ved et måltid. Ettersom det er den manuelle kalkulasjonen på for eksempel kalkulator, som tar mye tid, ville gruppen fjerne dette elementet for testpersonen. Dette gjøres nettopp ved å lagre matvarer, og den nødvendige informasjonen for å beregne insulinmengder. Matvarene skal videre bli presentert på en oversiktlig måte, og man skal kunne søke bland dem. Dette er et bevist valg for å redusere tiden testpersonen bruker, både i MVP-en og på å håndtere blodsukkeret generelt. Som nevnt, er det som regel kalkuleringen av insulinmengden som tar lengst tid. Ved at testpersonen i tillegg enkelt kan velge antall av matvarer fra en liste, det er ikke lenger behov for at testpersonen skal trenge å gjøre dette manuelt, noe som sparer mye tid.

Testpersonen har tatt i bruk to applikasjoner ved tidligere beregning av blodsukker, «Kalkulator» og «LibreLink» applikasjonene. LibreLink blir brukt ved å måle blodsukkerverdier til testpersonen i løpet av en dag. I tillegg loggfører testpersonen hvor mye insulin som blir satt til ulike måltider og til korrigerings. Gruppen så ikke en måte de kunne redusere tiden i denne applikasjonen. Grunnet dette fokuserte gruppen på å redusere tiden i Kalkulator applikasjonen. Ved å overføre alle beregningene til MVP-en, gjøre disse lettere ved at testpersonen plukker fra og velger antall fra en liste, antar gruppen at testpersonen ville spare tid ved reguleringen av blodsukkeret.

## 3.3 Utførelse

Det oppsto ulike utfordringer underveis i prosessen. For noen av disse utfordringene var det diverse tiltak som kunne løse problemet, mens andre utfordringer kun kunne forebygges.

### 3.3.1 Eksperiment prosess

Ettersom gruppen var avhengig av å samle inn nok data, ble det lagt vekt på å fullføre logging og registrering av insulin tidlig i prosessen, samt at algoritmen måtte ferdigstilles. Gruppen var klar over at diverse utfordringer kunne oppstå, som ville utsatt datainnsamlingen, eller minske den totale mengden data som var mulig å samle inn. Grunnet at flere av utfordringene ikke har noen tiltak når de først har inntruffet, for eksempel sykdom, arbeidet gruppen for å ferdigutvikle MVP-en for datainnsamling så tidlig som mulig. Dette var for å forebygge eventuelle tap i mengden data.

#### 3.3.1.1 Testperiode

Underveis i eksperimentet ble testpersonen syk, noe som førte til korrupt data. Denne dataen ble ansett som korrupt da insulin sensitiviteten under sykdom varierer sterkt fra friske dager, som nevnt i delsesjon 2.1. Ettersom rammeverket allerede var komplisert og stort nok å utvikle, valgte gruppen tidlig å ikke implementere «events» som blir nevnt i visjonsdokumentet, se vedlegg D. Dersom det hadde vært tid til å im-

---

plementere denne funksjonaliteten kunne dataen fra perioden testpersonen var syk blitt brukt i en annen sammenheng. Denne sammenhengen er da hvor mye insulin algoritmen skal beregne mens en person er syk. Etersom sykdom varierer fra person til person, både alvorlighet og lengde, konkluderte gruppen med at det ikke ville være mulig å samle inn nok data for at algoritmen skulle klare å beregne riktige mengder insulin for sykdom. Dette er også en av grunnene til at gruppen valgte å ikke implementere «events», da man ikke ville hatt mulighet til å teste denne funksjonaliteten ordentlig. Man kunne opparbeidet nok data på «treningsevent», men testpersonen trente ikke jevnlig under eller før testperioden startet, og det er også usikkert om man kunne samlet nok data på dette. Tilsammen ble ti dager med data tapt under perioden testpersonen var syk.

Senere under eksperimentet ble fem dager med data også tapt. Dette grunnet unormal kontinuerlig fysisk aktivitet, omtrent ti til tolv timer per dag. Som nevnt i delseksjon 2.1, påvirker fysisk aktivitet også insulinsensitiviteten. Etersom «events» ikke ble implementert, kunne ikke algoritmen ta høyde for denne type sensitivitet og insulinmengde. Disse datapunktene, som ved sykdom, ville vært ekstremalpunkter, og negativt påvirket algoritmens evne til å predikere riktig mengde insulin, uten spesialhåndtering.

Det var svært viktig at testpersonen kunne endre den estimerte insulinmengden til hva vedkommende faktisk ønsket å sette av insulin. En grunn til dette er at sprøytene til testpersonen har en presisjon på 0.5 enheter, mens estimatet fra algoritmen er på 0.1. Det vil derfor i nesten alle tilfeller være en liten avrundning som blir opp til testpersonen å ta stilling til. Grunnen til at gruppen valgte å ha en nøyaktighet på 0.1 i MVP-en, er for at testpersonen skal ha mer kontroll over avrundning, i tillegg til at det finnes for eksempel insulinpumper med 0.1 i presisjon.

En annen grunn til at det er viktig at testpersonen kan endre den estimerte insulinmengden, er at man må anta at algoritmen og estimatet ikke er fullstendig korrekt fra starten av, og trengs å tilpasses. Det vil derfor, spesielt i starten av testperioden, ikke være aktuelt for testpersonen å bare sette den foreslåtte mengden uten å verifisere den, da dette vil kunne føre til et dårligere regulert blodsukker, som igjen ikke er bra, se delseksjon 2.1. Derfor så gruppen det som nødvendig å la bruker endre den foreslåtte mengden, og loggføre mengden brukeren selv mente de trengte. MVP-en kan da i etterkant finne den teoretiske forskjellen denne endringen gjorde, og legge dette tilbake i feilen til estimatet. På denne måten kan bruker helt fritt selv velge dose uten at det påvirker læringen til algoritmen. Dette sikret også at testpersonen sin tidligere erfaring og intuisjon ikke blir et hinder for algoritmen sin læring. Et eksempel på dette er dersom testpersonen er vant til å sette fem enheter til to brødsiver, og MVP-en foreslår fire. I disse tilfellene vil testpersonen endre doseringen med en enhet (fra fire til fem). Om det fastende blodsukkeret etter måltidet ble det ønsket blodsukkeret, vil algoritmen «se» at den estimerte for lite insulin. Dette da det vil bli justert inn en teoretisk verdi i glukose (mmol/L) basert på «glukose-til-insulin»-forholdet, se 100-regelen i delseksjon 2.1, til testpersonen for den ekstra enheten med insulin testpersonen la inn. Dermed vil algoritmen anse dette som en høyere verdi enn det som faktisk ble oppnådd. Denne etterjusteringen er viktig da dette sikrer at algoritmen lærer av all dataen og testpersonen står fritt til å endre dosen.

Annet enn de tidligere nevnte utfordringene, var testpersonen god til å loggføre de nødvendige dataene, som blodsukkerverdi før og etter et måltid, mengde insulin satt til måltider og så videre. Dette gjorde at det kun var enkelttilfeller, som for eksempel

---

sykdom, som førte til at data gikk tapt.

### 3.3.1.2 Datainnsamling

Til tross for tiltakene for å innhente nok data, ble det fortsatt et tidspress på datainnsamlingen. Dette skyldes at gruppen ønsket å samle inn så mye data som mulig, og gjerne mer enn det som var planlagt i Gantt-diagrammet. I tillegg var det diverse bugs måtte fjernes fra koden før datainnsamlingen med algoritmen kunne begynne, slik at det ikke ble samlet inn ukorrekt data. Gruppen innså også at det ville være mange matvarer det ikke ville være mulig å samle inn nok data på, grunnet at frekvensen på denne matvaren var for lav. Til tross for dette, visste gruppen at det ville være mulig å samle inn nok data på dagsprofilene og enkelte matvarer som «brød».

### 3.3.1.3 Parametere til algoritmen

Gruppen fikk en idé i starten av perioden om å bruke simulasjon for å undersøke hvilke parametere til algoritmen som ga et best mulig resultat. Grunnet at dette ikke ble planlagt i Gantt-diagrammet, da idéen kom etter planleggingsperioden, og for lite tid og ressurser, måtte gruppen sløyfe denne idéen. Ettersom det var for lite tid til å gjennomføre og lage simulasjoner, valgte gruppen ut parameterene basert på teori og erfaringer.

En av parameterene til algoritmen er «minste antall tilfeller». Denne parameteren beskriver hvor mange datapunkter som må samles inn før algoritmen gjør en endring. Denne parameteren har blitt satt til ti, og det finnes flere grunner til dette. For det første undersøkte gruppen hvor mange datapunkter som er ideelt i ITS og segmentert lineær regresjonsanalyse. Fra forskningsartikkelen «*Segmented regression analysis of interrupted time series studies in medication use research*», står det at det er nødvendig med et tilstrekkelig antall punkter før og etter en avbrytelse av tidsrekken for å gjennomføre en segmentert regresjonsanalyse. I tillegg står det at en generell anbefaling er tolv datapunkter før og tolv datapunkter etter en slik avbrytelse [41]. For det andre undersøkte gruppen tidligere statistikk fra sensoren til testpersonen. Basert på disse dataene, valgte gruppen å ha ti tilfeller som minimumskrav før algoritmen kunne gjøre noen endringer. Dette er for at algoritmen skulle få samlet inn nok datapunkter til å gjøre gode endringer, uten at den viker for langt fra den generelle anbefalingen. En annen grunn til at ti tilfeller ble valgt, var fordi gruppen ville ha så mange iterasjoner av algoritmen som mulig, uten at det gikk ut over algoritmens evne til å gjennomføre gode endringer. Grunnen til at gruppen trengte at algoritmen gjennomførte nok iterasjoner, var den korte testperioden. For at algoritmen skulle ha noen praktisk nytte må den også være relativt rask til å tilpasse seg testpersonen, om «minste antall tilfeller» hadde vært 30, ville det kunne tatt flere år for enkelte matvarer å bli justert, så ti var et kompromiss mellom statistisk sikkerhet og praktisk brukbarhet/nytte. Basert på gjenværende tid for datainnsamlingen, tidligere sensordata, og den generelle anbefalingen om antall datapunkter, ble ti dager valgt som antall tilfeller/datapunkter før algoritmen kunne gjennomføre en endring.

Når algoritmen først utfører en analyse for å gjøre en endring, gjennomfører den en lineær regresjon av datapunktene. Deretter vil den flytte denne linjen så den ligger med den ene enden på x-aksen. Ettersom datapunktene er glukosefeil så vil

---

en linje på x-aksen tilsvare null i feil, altså det optimale tilfellet. Parameteren «nedre grense for prediksjonsintervall» forsikrer at prediksjonsintervallets nedre linje/bue ikke gir for lave verdier. Dette betyr at man ikke ønsker verdier under enn viss grense. Parameteren er beskrevet i mer detalj i delseksjon 3.2.5.

Den siste parameteren, «absolutt største justeringsfaktor», angir den største trygge endringen. Det er ikke ønskelig å gjennomføre for store endringer, grunnet den begrensede datamengden. Dette er fordi sannsynligheten for å få et dårlig datasett er større, som vil føre til at prediksjonsintervallet kan foreslå store endringer. Den «absolutt største justeringsfaktor» vil imidlertid kontrollere dette. Det vil si at man maksimalt kan endre/«flytte regressjonslinjen med én mmol/L» (glukoseverdi). Parameteren er beskrevet i mer detalj i delseksjon 3.2.5.

### 3.3.2 Resultatanalysen

Analysen av resultatet fra prosjektet ble gjennomført manuelt, etter at Python scriptene hadde blitt kjørt på testpersonen sin data. Dette var grunnet gruppens erfaringer med å analysere statistikk manuelt. Alle resultatene blir lagd basert på python-script, men selve evalueringen av disse blir gjennomført manuelt. Det er svært viktig å visualisere statistikken [2] for å kunne analysere trender og andre statistiske fenomen. Som nevnt i boken «*Statistikk for universiteter og høyskoler*» er det viktig å trekke mest mulig kunnskap ut av dataene i seg selv [2]. Det er kun når man ikke klarer å identifisere et mønster på basis av en visualisering av dataene at man tar i bruk mer avanserte analyseteknikker, som for eksempel maskinlæring. Gruppen så ikke behovet for mer avanserte teknikker, da de syntes analysen deres var tilstrekkelig.

## 3.4 Teknologi

### 3.4.1 Frontend-rammeverk

Ved valg av frontend-rammeverk ble det lagt stor vekt på at det skulle se estetisk pent ut med minimal innsats, og at rammeverket for datainnsamlingen, MVP-en, skulle kunne utvikles raskt. Dette er fordi det var essensielt å starte datainnsamlingen så tidlig som mulig, for å kunne ha resultater fra algoritmen å analysere. I starten av prosjektet ble det bestemt at MVP-en skulle være tilgjengelig på mobile enheter. Xamarin Forms er et cross-platform rammeverk som kan lage apper på iOS, Android og Windows. Et alternativ til Xamarin Forms var React Native. React Native er et Javascript rammeverk for å blant annet utvikle mobile applikasjoner for iOS og Android [42]. Det er basert på React, som brukes for utvikling mot nettlesere. Store deler av koden man skriver overføres direkte til andre plattformer [42], som gjør React Nativ ideelt for utvikling av både iOS og Android applikasjoner. Et av gruppens medlemmer hadde allerede god kjennskap til Xamarin Forms. Dersom man skulle tatt i bruk et annet rammeverk, ville begge gruppemedlemmene måtte lært seg dette fra bunnen av. Dette kunne resultert i at utviklingsprosessen tok svært lang tid, og eventuelt ført til dårlig kode, da ingen i gruppen hadde kjennskap til teknologien. Ettersom et av gruppemedlemmene allerede hadde kjennskap til Xamarin Forms, hadde tilgang på et kurs i dette, og Xamarin Forms sin evne til å skape estetisk pene applikasjoner på kort tid, ble dette frontend-rammeverket valgt. I tillegg var det spesifisert i oppgaveteksten at Xamarin Forms skulle brukes. Ettersom denne teksten ble utarbeidet

---

av gruppen selv, var det viktig for gruppen å begrunne hvorfor de valgte å spesifisere Xamarin Forms over for eksempel React Native.

### 3.4.2 Backend

Xamarin Forms har «sider» som kobles sammen, og består av et XAML og C# «code behind» par. Grunnet at Xamarin Forms var blitt valgt som frontend-rammeverk, benyttet derfor gruppen C# som kodespråk på backend. Gruppemedlemmene erfarte at C# minnet om allerede kjente kodespråk, og at det gikk raskt å anskaffe seg den nødvendige kunnskapen for å ta i bruk dette språket, og en av gruppens medlemmer hadde allerede kjennskap til språket. I tillegg sørget kurset som gruppen gjennomførte for at de fikk en innføring i C#. Et alternativ til C# hadde vært å bruke F# [43]. Grunnen til at F# ikke ble valgt er fordi gruppen hadde lite kjennskaper til det, og hadde allerede opparbeidet seg gode kunnskaper innen C# gjennom Xamarin Forms kurset.

#### 3.4.2.1 Databaser

For databaseløsningen valgte gruppen å bruke relasjonsdatabase system, SQLite. Grunnet valg av frontend-rammeverk, var det kun SQLite som Xamarin Forms støttet. Dette passet bra for gruppen, da SQLite er en relasjonsdatabase som gruppen ønsket å ta i bruk. I tillegg var det et ikke-funksjonelt krav beskrevet i visjonsdokumentet om å bruke en relasjonsdatabase. Grunnen til at gruppen ville bruke en relasjonsdatabase var at begge hadde tidligere kjennskap og erfaring til denne teknologien. Valget passet også problemstillingen meget godt. Ettersom gruppen var avhengige av å få rammeverket ferdigutviklet så raskt som mulig, valgte derfor gruppen noe de allerede hadde kjennskap til, i stedet for å bruke tid og ressurser på å lære noe nytt. Dette kunne også ført til en tregere prosess og flere feil. SQLite databasen tillater at Xamarin Forms applikasjoner laster og lagrer data lokalt på ulike enheter. SQLite er en serverløs databaseløsning, passer godt til mindre databaser, krever få konfigurasjoner for å settes opp og oppfylte de kriteriene gruppen trengte for databaseløsningen [44].

## 3.5 Arbeids- og rollefordeling

Det ble bestemt i starten av prosjektet at man skulle ha en jevn fordeling av arbeidsoppgaver, og Gantt-diagrammet ble flittig brukt for å holde en god oversikt over gruppens progressjon. Det var ingen faste roller under prosjektet. Ulike roller ble heller dynamisk fordelt etter behov. I starten av prosjektet gjennomførte gruppens medlemmer et teoretisk og praktisk kurs, slik at begge skulle være i stand til å gjøre alle de ulike oppgavene. I tillegg hadde begge medlemmene en god innsikt i diabetes type 1 fra tidligere informasjonsinnhenting og erfaringer. Under MVP-ens utvikling, samarbeidet gruppens medlemmer gjennom parprogrammering, se delseksjon 3.5.1. Etter at MVP-en og algoritmen var ferdig utviklet, og det kun var opprydding i koden og tester som gjensto, ble det vurdert at kun én person trengte å arbeide med dette av gangen. Gruppen anså det som mer hensiktsmessig å fordele arbeidet på dette punktet, da det var andre arbeidsoppgaver som også måtte gjøres. Dette var for eksempel dokumentasjon, og gjennomføringen av brukertester. Fra dette punktet arbeidet

---

gruppemedlemmene mer adskilt, men progresjonen på de ulike arbeidsoppgavene ble delt på slutten av hver arbeidsdag, gjennom et daglig oppsummeringsmøte. På denne måten hadde begge en oversikt over prosjektets progresjon i sin helhet. Dette sørget for en god flyt i arbeidet, i tillegg til at man kunne gjøre justeringer underveis. Disse justeringene kunne være at for eksempel begge medlemmene arbeidet på en oppgave som tok lenger tid enn forventet.

### **3.5.1 Parprogrammering**

Ved prosjektets start ble det bestemt at gruppen skulle ta i bruk parprogrammering for utvikling av MVP-en for datainnsamlingen. Grunnen til dette var at man på denne måten kunne ha kontinuerlig code review som forebygger feil og dårlig kode [34]. Parprogrammering er en teknikk hentet fra agile arbeidsmetoder, og den blir blant annet brukt i Extreme Programming. I tillegg har gruppen hatt god erfaring med denne arbeidsmetoden tidligere, da den åpner for kontinuerlig diskusjon, og man får tenkt gjennom ulike problemstillinger som man kunne ha oversett alene. Det var også flere ulike aspekter ved MVP-en som måtte bli diskutert fortløpende etter hvert som ulike problemer og beslutninger måtte tas. I disse tilfellene var det svært praktisk med parprogrammering, og gruppen følte det var en god utnyttelse av ressursene. Grunnen til dette er fordi man sparte tid på å kontinuerlig diskutere og ta beslutninger, i stedet for å avbryte andre arbeidsoppgaver for å ta disse beslutningene gjentatte ganger. En ulempe ved parprogrammering er at det til tider ikke oppleves som den beste måten å utnytte ressurser og tid på. Dette tilfellet oppsto imidlertid ikke under utviklingen. Dersom dette hadde oppstått, hadde gruppen planlagt å fordele arbeidet til det ble fornuftig med parprogrammering igjen. Et eksempel der parprogrammering i høyeste grad var viktig og fornuftig, var under utviklingen av algoritmen. Som nevnt i delseksjon 2.1, kan feil insulinmengde medføre fatale konsekvenser for diabetikere. Gjennom parprogrammeringer passet gruppen på å ha en kontinuerlig diskusjon under utviklingen av algoritmen, for å sørge for at denne ble implementert riktig, slik at det ikke skulle være noen uventede konsekvenser. Gruppen fikk dermed implementert algoritmen med kontinuerlig code review, som forebygger bug-fiksing senere i prosessen, som igjen sparer tid, samt at gruppen fikk diskutert den beste løsningen og implementasjonen av algoritmen og medlemmene fikk trukket frem ulike problemstillinger knyttet til denne.

## **3.6 Statistikk**

Ved oppgavetekstens utforming ble det besluttet at man skulle ta i bruk statistikk for analysen av den beregnede insulinmengden, og at denne analysen skulle foregå manuelt. Grunnet at statistikk var et emne begge medlemmene av gruppen har gode forkunnskaper i, og dens deterministiske natur, ble statistikk valgt for vurdering av algoritmens analyse.

### **3.6.1 Interrupted time series**

For å analysere dataene tok gruppen i bruk ITS. Som beskrevet i delseksjon 2.2.11, gjør ITS at man kan analysere hvordan en variabel utvikler seg over tid, og for å etablere en underliggende trend, som blir avbrutt av en intervensjon på et kjent tid-

---

spunkt. Når algoritmen får inn nok data på for eksempel en matvare, vil den undersøke om den skal endre skalaren til matvaren. Dersom den velger å endre skalaren, vil dette være der tidsrekken blir avbrutt, og dermed være det kjente tidspunktet. For skalarendringer forventer gruppen en momentan endring som kan observeres med en gang. Derfor brukes ITS modellen 16 uten  $\beta_3$ -leddet som beskrevet i delseksjon 2.2.11. For analysen av sensor dataene fra LibreLink var det forventet et utslag med en gang, men også en endring med tiden da algoritmen begynte å endre verdier. For disse dataene valgte gruppen å bruke den fullstendige ligningen 16 for å modellere dataene.

Det ble diskutert ulike metoder for å analysere resultatet av algoritmen. Blant annet ble enkel lineær regresjon, multippel lineær regresjon, og tidsrekker undersøkt. Enkel lineær regresjonsanalysen ble vurdert for å undersøke om det var en tydelig trend, for eksempel en synkende trend dersom blodsukkeret generelt har vært for høyt. Grunnen til at denne ble valgt bort, var at den ikke kunne ta hensyn til algoritmens endringer på samme måte som ITS, og hvordan dette påvirker modellene.

Multippel lineær regresjon ble vurdert ettersom det finnes flere mulige forklaringsvariabler for insulinmengden på et gitt tidspunkt. For eksempel kan mengden insulin avhenge av alle faktorene trening, sykdom, tidspunkt på dagen og så videre. Multippel lineær regresjon vil hjelpe med å finne den beste lineære sammenhengen mellom de ulike forklaringsvariablene og responsen [2]. Grunnen til at multippel lineær regresjon ble valgt bort, var fordi det ville tatt for lang tid å implementere funksjonaliteten i MVP-en. Dette er da funksjonaliteten for «events» som beskrevet i visjonsdokumentet. I tillegg ville heller ikke denne analysen tatt hensyn til algoritmens endringer slik som ITS.

Tidsrekkeanalyse er en teknikk som gruppen vurderte da den brukes for å analysere hvordan en variabel utvikler seg med tiden, dersom målinger blir tatt med jevne tidsintervaller [2]. Tidsrekkemodeller er best egnet til å beskrive naturlige systemer, som for eksempel temperaturvariasjon, mens de er mindre gode på å predikere fenomener som menneskelig atferd. En grunn til at denne metoden ble valgt bort er fordi ITS var et bedre alternativ. Dette er fordi ITS gjør nøyaktig hva tidsrekker er i stand til, nemlig predikere fremtiden, men ITS tar i tillegg hensyn til algoritmens endringer.

Ettersom ITS er godt egnet for intervensjoner introdusert på et gitt tidspunkt, og grunnet tidsrestriksjon, konkluderte gruppen med at ITS gir gode grafiske resultater som gjør analyseringen lettere, og at denne analysen var den mest egnede for oppgaven.

## **3.7 Algoritmen**

### **3.7.1 Konstruksjon**

Under algoritmens konstruksjon anskaffet gruppen seg en felles forståelse av problemet. Det har blitt utviklet et bokføringsverktøy, MVP-en, som loggfører hva og når en person spiser, hvor mye insulin de setter til dette, hvor mye de teoretisk skal sette, hva glukose-verdien deres er før måltidet, og hva den er ønsket å være etter måltidet. Alt dette er en del av MVP-en uten algoritmen. Gruppen hadde dette som grunnlag for hvilke data som var tilgjengelig. Målet var å lage en algoritme som brukte disse dataene i en form for analyse, for å kunne justere verdiene for matvarer og dagsprofiler. Dette for å få en bedre estimert insulin mengde fra kalkulatoren.



---

Gruppen ble enig om å ha «skalarer» koblet til alle matvarer. I de tilfellene algoritmen mente en matvare trengte mer eller mindre insulin, kunne den stille matvarens skalar henholdsvis opp eller ned, uten å endre matvarens opprinnelige verdier. Skalarene begynner med en verdi på én, og på denne måten starter man med den opprinnelige teoretiske verdien. Grunnen til at hver matvare får sin egen skalar, er for å håndtere de forskjellige typene karbohydrater matvarene kan ha på en enkel måte. Dette er for å kunne ha en innvirkning på hvordan blodsukkeret endres, grunnet forskjellen i hvordan kroppen fordøyer de ulike karbohydratene [45]. Det ble også laget en skalar som skulle ta hånd om tilfeller hvor det kun ble satt korrigeringsinsulin, dette da testpersonen vanligvis satte mindre i disse tilfellene.

For dagsprofiler ble gruppen enig om å ha to skalarer, én til insulinet for karbohydratene i måltidet, og én til insulinet for korrigerende glukose til måltidet. Gruppen var usikre på om forholdet mellom disse var sterkt, men anså det som positivt å tillate at verdiene kunne stilles individuelt. Dersom de henger sterkt sammen, vil de endres relativt likt uansett og derav vil ikke valget av å ha to skalarer endre resultatet negativt. Bakgrunnen til disse skalarene er at de skal ta hånd om variasjonen i insulin-effektiviteten i kroppen gjennom dagen, som for eksempel «Dawnfenomenet», se delseksjon 2.1.

Gruppen valgte i tillegg å koble «glukose-til-insulin»-forholdet opp mot en skalar. Tanken var at den opprinnelige sensitiviteten bruker legger inn første gang i MVP-en kunne endre seg med tiden. Derfor ble det også knyttet en skalar opp mot denne, for å redegjøre for endringer i sensitiviteten.

For at algoritmen skulle kunne evaluere hvor godt insulinestimaten til, for eksempel et måltid var, var den nødt til å ha en verdi som sier hvor godt estimaten var. Gruppen løste dette ved at et hvert loggført måltid blir koblet opp mot en påminnelse. Denne påminnelsen vil etter tre timer spørre om den oppnådde glukoseverdien ved dette tidspunktet. Algoritmen ser på differansen mellom oppnådd verdi og ønsket verdi, og gjør endringer for å forsøke å få differansen til å bli null.

Gruppen vurderte, og begynte, å implementere en annen løsning hvor «insulin-til-glukose»-forholdet ble oppdatert basert på TDD gjennomsnittet over de siste sju dagene. Dette ved å bruke TDD gjennomsnittet og 100-regelen, se delseksjon 2.1, men gruppen fant for mange fallgruver ved mangel på innlagt data, og annet som kunne forskyve «insulin-til-glukose»-forholdet drastisk. Gruppen valgte heller å bruke en skalar, ettersom det allerede var laget sikkerhetsmekanismer gjennom algoritmen, se delseksjon 3.2.5, som beskyttet mot de identifiserte problemene ved den andre løsningen.

Etter at gruppen hadde kartlagt hva slags data algoritmen behøvde, og hva den skulle gjøre som respons på den gitte dataen, kunne medlemmene utarbeide hvordan algoritmen skulle operere internt. Gruppen ble enige om hvilke komponenter algoritmen skulle bestå av, og det ble deretter utarbeidet en plan for hvordan algoritmen skulle operere. Algoritmen ble også visualisert, se vedlegg B. I tillegg ble det planlagt i hvilken rekkefølge komponentene skulle implementeres. Denne rekkefølgen ble justert etterhvert som gruppen så behov for dette. Deretter implementerte gruppen de nødvendige komponentene i MVP-en. Gruppen definerte når algoritmen skulle bli utført, og begynte deretter å implementere førsteutkastet til algoritmen. For å sikre integriteten til algoritmen, forsikret gruppen seg først og fremst om at koden til algoritmen hadde de nødvendige sikkerhetsmekanismene på plass og at koden hadde

---

god unntakshåndtering.

### 3.7.2 Virkemåte

---

#### Algorithm 1 Main algorithm

---

```
1: Input : Reminder
2: MINIMUM_OCCURENCES  $\leftarrow$  10
3: LOWER_BOUND_FOR_PREDICTION_INTERVAL  $\leftarrow$  -1.0
4: ABSOLUTE_MAXIMUM_DISTANCE_CHANGE  $\leftarrow$  1.0
5: Logs  $\leftarrow$  PartitionGlucoseErrorToLogs(Reminder)
6: UpdateScalarValues(Logs)
```

---

Algoritmen starter med at en bruker legger inn en glukoseverdi i en påminnelse, kalt «Reminder» i kildekode. Det vil si at et måltid blir ansett som ferdig, og den faktisk oppnådde glukoseverdien er tilgjengelig for å finne differansen til den ønskede verdien. Den overordnede algoritmen vises i algoritme 1.

---

#### Algorithm 2 *PartitionGlucoseErrorToLogs*(*Reminder*)

---

```
1: Input : Reminder
2: Logs  $\leftarrow$  Reminder.Logs
3: TargetGlucoseValue  $\leftarrow$  Logs.Last.DayProfile.TargetGlucose
4: GlucoseErrorInReminder  $\leftarrow$  Reminder.GlucoseAfterMeal - TargetGlucoseValue
5: TotalInsulinGiven  $\leftarrow$  Sum(Logs.InsulinEstimate)
6: for Log in Logs do
7:   TargetGlucoseForLog  $\leftarrow$  Log.DayProfile.TargetGlucoseValue
8:   GlucoseErrorForLogPartitioned  $\leftarrow$  GlucoseErrorInReminder *  $\frac{\textit{Log.InsulinEstimate}}{\textit{TotalInsulinGiven}}$ 
9:
10:  GlucoseErrorAdjusted  $\leftarrow$  GlucoseErrorForLogPartitioned - (Log.InsulinEstimate - Log.InsulinFromUser) * Global.InsulinToGlucoseRatio
11:
12:  Log.GlucoseAfterMeal  $\leftarrow$  TargetGlucoseForLog + GlucoseErrorAdjusted
13: end for
14: return Logs
```

---

Deretter vil algoritmen ta påminnelsen, og finne differansen mellom den oppnådde glukoseverdien og den ønskede verdien i den siste loggen (Log) koblet til påminnelsen. Denne differansen, eller feilen, i glukose vil bli fordelt utover alle loggene som er koblet til påminnelsen. Dette vil si at det foregår en fordeling av ansvar for feilen. Partisjoneringen av differansen mellom loggene er basert på hver logg sin andel av det totale bidraget av insulin. Dette vil si at dersom en påminnelse hadde tre logger koblet til seg og den ene sto for 50% av insulinet de tre loggene har til sammen, vil denne loggen få ansvar for 50% av differansen, se algoritme 2. Ofte vil bruker ha endret doseringen som ble foreslått, og dette må korrigeres for. Dette gjøres ved at differansen mellom estimatet og den faktiske satte verdien, konverteres fra insulin til glukose ved «insulin-til-glukose»-forholdet, se 100-regelen i delseksjon 2.1. Den teoretiske endringen i glukose, som følge av bruker sin endring av dosering, blir lagt sammen med differansen loggen fikk tildelt fra påminnelsen. Denne verdien, sammenlagt med den ønskede glukoseverdien for loggen, vil være den justerte «glukose-etter-måltid»-verdien (*GlucoseAfterMeal*) i loggen, se algoritme 2.

---

**Algorithm 3** UpdateScalarValues(Logs)

---

```
1: Input : Logs
2: for Log in Logs do
3:   UpdateDayProfileScalars(Log.DayProfile)
4:   UpdateCorrectionInsulinScalar()
5:   for Grocery in Log.Groceries do
6:     UpdateGroceryScalar(Grocery)
7:   end for
8: end for
```

---

Når partisjoneringen til loggene er ferdig, vil algoritmen undersøke alle elementer involvert i påminnelsen. Den vil sjekke alle dagsprofiler (DayProfile) og matvarer (Grocery) i alle loggene koblet til påminnelsen, i tillegg til korrigeringsinsulinet (CorrectionInsulin), se algoritme 3.

---

**Algorithm 4** GetDataPointForDayProfileCarbScalar(Log)

---

```
1: Input : Log
2:  $GlucoseErrorForCarbsPartitioned \leftarrow Log.GlucoseError * \frac{Log.InsulinForCarbs}{Log.InsulinEstimate}$ 
3:
4: return  $DataPoint(Log.Date, GlucoseErrorForCarbsPartitioned)$ 
```

---

---

**Algorithm 5** GetDataPointForDayProfileGlucoseScalar(Log)

---

```
1: Input : Log
2:  $GlucoseErrorForGlucosePartitioned \leftarrow Log.GlucoseError * \frac{Log.CorrectionInsulin}{Log.InsulinEstimate}$ 
3:
4: return  $DataPoint(Log.Date, GlucoseErrorForGlucosePartitioned)$ 
```

---

---

**Algorithm 6** UpdateDayProfileScalars(DayProfile)

---

```
1: Input : DayProfile
2:
3: LogsWithDayProfile  $\leftarrow$  Database.GetAllLogsWithDayProfile(DayProfile)
4:
5: CarbScalar  $\leftarrow$  Database.GetCarbScalarForDayProfile(DayProfile)
6: GlucoseScalar  $\leftarrow$  Database.GetGlucoseScalarForDayProfile(DayProfile)
7:
8: LogsForCarbScalar  $\leftarrow$  Filter(LogsWithDayProfile where Log.Date is after CarbScalar.Date)
9: LogsForGlucoseScalar  $\leftarrow$  Filter(LogsWithDayProfile where Log.Date is after GlucoseScalar.Date)
10:
11: DataPointsForCarbScalar  $\leftarrow$  Empty list
12: for Log in LogsForCarbScalar do
13:   DataPointsForCarbScalar.Add(GetDataPointForDayProfileCarbScalar(Log))
14: end for
15: DataPointsForGlucoseScalar  $\leftarrow$  Empty list
16: for Log in LogsForGlucoseScalar do
17:   DataPointsForGlucoseScalar.Add(GetDataPointForDayProfileGlucoseScalar(Log))
18: end for
19: if DataPointsForCarbScalar.Size  $\geq$  MINIMUM_OCCURENCES then
20:   Distance  $\leftarrow$  GetGreatestSafeDistanceFromWantedLine(DataPointsForCarbScalar)
21:
22:   InsulinGivenOnAvgForCarbs  $\leftarrow$  LogsForCarbScalar.Sum( $\frac{\text{Log.GetInsulinForCarbs}}{\text{logsForCarbScalar.Size}}$ )
23:
24:   ExtraInsulin  $\leftarrow$   $\frac{\text{Distance}}{\text{Global.InsulinToGlucoseRatio}}$ 
25:
26:   ScaleFactor  $\leftarrow$   $\frac{\text{InsulinGivenOnAvgForCarbs} + \text{ExtraInsulin}}{\text{InsulinGivenOnAvgForCarbs}}$ 
27:
28:   CarbScalar  $\leftarrow$  CarbScalar * ScaleFactor
29: end if
30:
31: if DataPointsForGlucoseScalar.Size  $\geq$  MINIMUM_OCCURENCES then
32:   Distance  $\leftarrow$  GetGreatestSafeDistanceFromWantedLine(DataPointsForGlucoseScalar)
33:
34:   InsulinGivenOnAvgForGlucose  $\leftarrow$  LogsForGlucoseScalar.Sum( $\frac{\text{Log.CorrectInsulin}}{\text{logsForGlucoseScalar.Size}}$ )
35:
36:   ExtraInsulin  $\leftarrow$   $\frac{\text{Distance}}{\text{Global.InsulinToGlucoseRatio}}$ 
37:
38:   ScaleFactor  $\leftarrow$   $\frac{\text{InsulinGivenOnAvgForGlucose} + \text{ExtraInsulin}}{\text{InsulinGivenOnAvgForGlucose}}$ 
39:
40:   GlucoseScalar  $\leftarrow$  GlucoseScalar * ScaleFactor
41: end if
```

---

Det som skjer når algoritmen undersøker en dagsprofil, er at den vil hente ut alle logger med den gitte dagsprofilen som har blitt lagt inn, siden forrige endring av skalarene i dagsprofilen, se algoritme 6. Dersom det er nok slike logger, flere enn minstekravet, se delseksjon 3.3.1.3, kan den gjøre en endring, og vil først hente ut datapunkter fra loggene. Disse datapunktene består av datoen og tiden loggen ble laget, som X-verdier. Y-verdiene blir hentet ut som vist i algoritmene 4 og 5. Merk at det vil lages to lister med datapunkter, én for karbohydrat-skalaren (*CarbScalar*) i dagsprofilen, og én for glukose-skalaren (*GlucoseScalar*).

Det er først når datapunktene er generert at en analyse kan kjøres, og denne vil foreslå en endring i skalar-verdien. Denne analysen blir beskrevet i mer detalj i senere avsnitt. Distansen analysen returnerer, er en endring i Y-aksen, altså en en-

dring i glukose, med enhet «mmol/L». Denne endringen blir gjort om til enheter med insulin, ved «insulin-til-glukose»-forholdet som nevnt tidligere. Insulinmengden representerer endringen i mengde insulin, som vil føre til at fremtidige glukoseverdier blir bedre. Man finner en skaleringsfaktor som vist i algoritme 6. Denne skaleringsfaktoren vil bli multiplisert inn i de respektive skalarene, for karbohydrater og glukose, se algoritme 6, slik at fremtidige doseringer vil bruke de nye verdiene.

---

#### Algorithm 7 UpdateGroceryScalar(Grocery)

---

```

1: Input : Grocery
2:
3: GroceryScalar ← Database.GetGroceryScalar(Grocery)
4: LogsAfterDate ← Database.GetAllLogsAfterDate(GroceryScalar.Date)
5: LogsForGroceryScalar ← Filter(LogsAfterDate where Log.Groceries.Contains(Grocery))
6:
7: DataPointsForGroceryScalar ← Empty list
8: for Log in LogsForGroceryScalar do
9:   DataPointsForGroceryScalar.Add(GetDataPointForGroceryScalar(Log, Grocery))
10: end for
11:
12: if DataPointsForGroceryScalar.Size ≥ MINIMUM_OCCURENCES then
13:   Distance ← GetGreatestSafeDistanceFromWantedLine(DataPointsForGroceryScalar)
14:
15:   ExtraInsulin ←  $\frac{Distance}{Global.InsulinToGlucoseRatio}$ 
16:
17:   ScaleFactor ←  $\frac{Grocery.InsulinPerPortion + ExtraInsulin}{Grocery.InsulinPerPortion}$ 
18:
19:   GroceryScalar ← GroceryScalar * ScaleFactor
20: end if

```

---



---

#### Algorithm 8 GetDataPointForGroceryScalar(Log, Grocery)

---

```

1: Input : Log
2: GroceryGlucoseError ←  $\frac{Log.GlucoseError * Grocery.InsulinPerPortion}{Log.InsulinEstimate}$ 
3:
4: return DataPoint(Log.Date, GroceryGlucoseError)

```

---

Det er en relativt lik prosess for matvarer som for dagsprofiler. Forskjellen mellom de to er at det kun er en karbohydrat-skalar for matvarer. Logger som involverer matvaren, og som er lagt inn etter forrige skalarendring, hentes ut, se algoritme 7. Dersom det er nok logger, blir det generert datapunkter. Datapunktene for matvarer bruker også dato og tid fra loggen som X-verdi, men for Y-verdiene, brukes andelen av feilen i loggen basert på bidraget av insulin matvaren hadde per porsjon, se algoritme 8. Det vil bli kjørt samme analyse som på dagsprofiler. Her vil også det teoretiske ekstra insulinet bli beregnet basert på distansen fra analysen delt på «insulin-til-glukose»-forholdet. En skalarfaktor vil også bli beregnet i dette tilfellet, men ved bruk av insulin per porsjon for matvaren, og ikke gjennomsnittlig insulin for dagsprofilen. Til slutt vil matvarens skalar oppdateres ved å gangens med skaleringsfaktoren, se algoritme 7.

---

**Algorithm 9** UpdateCorrectionInsulinScalar()

---

```
1: CorrectionScalar ← Database.GetCorrectionScalar()
2: LogsAfterDate ← Database.GetAllLogsAfterDate(CorrectionScalar.Date)
3: DataPointsForCorrectionScalar ← Empty list
4: for Log in LogsAfterDate do
5:   DataPointsForCorrectionScalar.Add(GetDataPointForCorrectionScalar(Log))
6: end for
7:
8: if DataPointsForCorrectionScalar.Size ≥ MINIMUM_OCCURENCES then
9:   Distance ← GetGreatestSafeDistanceFromWantedLine(DataPointsForCorrectionScalar)
10:
11:    $ExtraInsulin \leftarrow \frac{Distance}{Global.InsulinToGlucoseRatio}$ 
12:
13:   NewScalar ← 1 + ExtraInsulin
14:   CorrectionScalar ← NewScalar
15:
16:    $Global.InsulinToGlucoseRatio \leftarrow \frac{Global.InsulinToGlucoseRatio}{CorrectionScalar}$ 
17: end if
```

---

---

**Algorithm 10** GetDataPointForCorrectionScalar(Log)

---

```
1: Input : Log
2:  $GlucoseErrorPerUnitOfCorrectionInsulin \leftarrow \frac{Log.GlucoseError}{Log.InsulinEstimate}$ 
3:
4: return DataPoint(Log.Date, GlucoseErrorPerUnitOfCorrectionInsulin)
```

---

Algoritmen sjekker også korrigerings insulinet, denne er også relativt lik dagsprofilen og matvaren. Den henter alle logger som har korrigerings insulin og som er lagt inn etter siste skalar endring, se algoritme 9. Igjen om det er nok logger så vil det bli generert en liste med datapunkter. I dette tilfellet er også X-verdien dato og tid fra loggen, men Y-verdien er loggens glukose feil delt på loggens insulin estimat, som gir feilen per enhet av korrigerings insulin, se algoritme 10.

Den samme analysen vil også bli kjørt på datapunktene, og en distanse vil bli returnert. Ekstra insulin vil bli kalkulert på samme måte, ved at distansen blir delt på «insulin-til-glukose»-forholdet. Den nye skalaren her vil være: én addert med det ekstra insulinet. Dette da man kan se på feil per enhet med korrigeringsinsulin. Korrigeringskalaren vil bli satt lik denne nye verdien, og «insulin-til-glukose»-forholdet vil bli oppdatert til seg selv dividert med den nye korrigeringsinsulinskalaren.

---

**Algorithm 11** GetGreatestSafeDistanceFromWantedLine(DataPoints)

---

```
1: Input : DataPoints
2:
3:  $X_{values} \leftarrow DataPoints.Date$ 
4:  $Y_{values} \leftarrow DataPoints.GlucoseError$ 
5:
6:  $Alpha, Beta \leftarrow LinearRegression(X_{values}, Y_{values})$ 
7:  $UpperBound, LowerBound \leftarrow PredictionInterval(X_{values}, Y_{values}, Alpha, Beta, 95\%)$ 
8:
9:  $SmallestDifferenceToXAxis \leftarrow Min(Alpha + Beta * X_{values}.First, Alpha + Beta * X_{values}.Last)$ 
10:  $Distance \leftarrow Min(SmallestDifferenceToXAxis, LowerBound - LOWER\_BOUND\_FOR\_PREDICTION\_INTERVAL)$ 
11: if  $Abs(Distance) > ABSOLUTE\_MAXIMUM\_DISTANCE\_CHANGE$  then
12:   return  $\frac{Distance * ABSOLUTE\_MAXIMUM\_DISTANCE\_CHANGE}{Abs(Distance)}$ 
13: end if
14: return  $Distance$ 
```

---

Selve analysen som blir utført på datapunkter, er som sagt lik i tilfellene nevnt over. Først vil datapunktene bli delt inn i to lister, én for X-verdier, datoene fra loggene, og én til Y-verdier, glukosefeilen. Det vil bli kjørt en enkel lineær regresjon på disse verdiene, som gir en alpha og beta verdi. Deretter sendes X- og Y-verdiene sammen med alpha og beta, og til slutt en sikkerhet på 95%. Dette utgjør et 95% prediksjonsintervall for datapunktene. Fra dette får man en øvre og nedre predikert verdi for det siste punktet langs X-aksen, da dette representerer best det neste forventede punktet. Til slutt finner analysen den lengste distansen som er tillatt av restriksjonen fra sikkerhetsmekanismene, se delseksjon 3.2.5. Man finner den minste avstand fra regresjonslinjen til X-aksen, da man ønsker at linjen skal ligge på X-aksen som tilsvarer null glukosefeil. Så sjekker den hvor mye prediksjonsintervallets nedre grense tillatter å bli flyttet. Dersom dette er mindre enn avstanden til regresjonslinjen, vil prediksjonsintervallets nedre grense overkjøre regresjonslinjen. Avslutningsvis vil man begrense den totale endringen. Dersom endringen som analysen ønsker å gjennomføre er større enn grensen tillatt, vil man benytte seg av den maksimale mulige endringen, se algoritme 11. Det er denne sluttverdien som blir returnert, og som er distansen i de tidligere nevnte tilfellene over.

### 3.7.3 Alternativ

Et alternativ til å konstruere algoritmen hadde vært å ta i bruk maskinlæring. Dette ble vurdert, men gruppen konkluderte med at det var for mange fallgruver med maskinlæring. Blant disse finner man blant annet «black-box». En del maskinlæringsalgoritmer er i praksis uforståelige. Dette vil si at ens egen intuisjon ikke klarer å fortolke prediksjonsmodellen. Man forstår altså ikke hva slags logisk sammenheng det er mellom input og output, og man forstår heller ikke hva som foregår i modellen. Selv om modellen kan gi gode prediksjoner basert på enkeltdata, kan den være mindre god til å gi generell informasjon. Et eksempel er klassifisering av en husky versus ulv basert på bilder. Her tok maskinlæringsalgoritmen kun hensyn til om det var snø i bildet. Dette resulterte i at den ga en god nok klassifisering, selv om bildegjenkjenningen i seg selv var mer tvilsom [2]. Andre fallgruver ved maskinlæring er blant annet feiltolkning av kvalitative data, overtilpasning, årsak/virkning, og stokastikk versus determinisme. Årsak og virkning fallgruven er i likhet med vanlig regresjons- og kor-

---

relasjonsanalyse, der maskinlæringsalgoritmen ikke sier noe om hva som er årsak og hva som er virkning. Med stokastikk versus determinisme handler om at ikke alt kan beregnes og forutsies. Man kan ikke spå et utfall med 100% sikkerhet, uansett hvor mye data som er samlet inn, eller hvor gode modeller man har. Ettersom MVP-en er ment som et medisinsk verktøy, er det svært viktig for gruppen at den ble utviklet med teknologi som begge gruppens medlemmer forsto. Altså at man ikke ville fått et tilfelle slik som med «black-box». Dette er fordi det skal svært lite feilberegning til før insulinmengder kan gi alvorlige konsekvenser. Grunnet den begrensede mengden data var det heller ikke mulig å ta i bruk maskinlæring da dette krever større mengder data. Gruppen tror ikke MVP-en ville generert nok data gjennom testperioden for å kunne trent en maskinlæringsmodell. Hvert fall ikke hatt nok tid igjen til å se om denne modellen fungerte. Det er også vanskeligere å ha en dynamisk kontinuerlig lærende modell med maskinlæring. Ved en algoritme så har man mer determinisme og kan legge inn veldig spesifikke restriksjoner, se delseksjon 3.2.5. Dette skaper trygghet rundt algoritmen og valgene den gjør kontra en «black-box». I algoritmen er det mulig å utnytte nyttig innsikt i diabetes, som en sykdom og hvordan den fungerer, for å hente ut mer informasjon fra den begrensede mengden data.

### 3.8 Brukertester

Det ble gjennomført to brukertester, og de ble gjennomført både på en diabetiker og en person uten diabetes. Den ene brukertesten ble gjennomført digitalt, og var semi-strukturert. Grunnen til at brukertesten ble gjennomført digitalt var at personen som skulle gjennomføre brukertesten og testerene oppbefant seg på ulike steder, og hadde ikke mulighet til å treffes fysisk. I dette tilfellet var det ingen andre alternativer, og testen ble gjennomført via skjermdeling av MVP-en. Den andre brukertesten ble gjennomført fysisk, og var også semi-strukturert.

Brukertestene var semi-strukturert for å undersøke hvor intuitiv MVP-en er i seg selv, ved å la brukeren selv teste ut funksjonaliteten, men samtidig få testet all funksjonalitet gjennom åpne spørsmål uten å lede brukeren for mye. De andre alternativene til semi-strukturert brukertest er da strukturert og ustrukturerte tester. Fordelen med strukturerte brukertester er at absolutt all funksjonalitet kan testes i den forventede rekkefølgen som testerene har planlagt. Ulempen er at det ikke gir brukeren frihet til å teste MVP-en på egenhånd, og dermed får man ikke testet i like stor grad om designet er intuitivt uten åpne spørsmål. I tillegg kan man lede brukeren for mye gjennom testen. For ustrukturerte brukertester har man da fordelen av at man får testet i stor grad hvor intuitivt designet er, mens ulempen er at deler av funksjonaliteten ikke blir testet, og man har ingen kontroll over testingen. Ved en semi-strukturert brukertest vil ulempen være at det blir utfordrende for testerene å holde oversikt over testingen, i tillegg til at det blir vanskeligere å sammenligne brukere. Ettersom gruppen kun testet på kun to personer, der kun én hadde diabetes, annet enn testpersonen gjennom hele prosessen, var ikke dette et problem. Dette var fordi formålet med brukertestene var ulikt. For diabetikeren ønsket gruppen å få testet hele MVP-en, og undersøke om funksjonaliteten gir mening for andre diabetikere enn testpersonen. Ved brukertesten med en ikke-diabetiker, var formålet kun å få et generelt inntrykk av hvor intuitivt designet var. Semi-strukturerte brukertester gir også brukeren mulighet til å oppføre seg så normalt som mulig, da brukeren får oppgaver som skal løses, uten å få klare retningslinjer på hvordan [36].



---

Etter den første brukertesten ble gjennomført, erfarte gruppen at det var lurt å ha eksisterende data i databasen, slik at bruker hadde mulighet til å utforske MVP-en mer på egenhånd. Dette var for at bruker ikke måtte finne ut av at de måtte opprette en matvare og dagsprofil før de kunne bruke resten av MVP-en.

### **3.9 Måloppnåelse**

For å nå både resultat- og effektmålene beskrevet i forprosjektsplanen, ble det gjennomført ulike tester for å sørge for at disse målene kunne testes, og dermed gi et målbart resultat.

#### **3.9.1 Resultatmål**

For å nå resultatmålene sørget gruppen for å samle inn den nødvendige dataen på begynnelsen av prosjektet, gjennom prosjektet, og ved prosjektets slutt. Dataen som ble samlet inn var blant annet hvor mye tid testpersonen brukte på å regulere blodsukkeret i ukene før og etter bruk av MVP-en. I tillegg ble det samlet inn data om HbA1c-verdien kontinuerlig, både før og gjennom perioden med MVP-en i bruk. Dette ble samlet inn gjennom testpersonen sin sensor applikasjon, som måler denne verdien og gir et estimat på denne hver andre uke. Dette var for å kunne se hva slags effekt MVP-en hadde, og for å teste målene.

Dataen for tidsbruk ble samlet inn basert på skjermtiden logget av testpersonens telefon på applikasjonene «Kalkulator» og «LibreLink», se delseksjon 2.6. Testpersonen var nøye med å ikke bruke «Kalkulator»-applikasjonen for utregning av noe annet enn blodsukker i perioden for innsamlingen av data, fire uker, for å få et så korrekt estimat som mulig. For å samle inn dataen på hvor lang tid testpersonen brukte på å regulere blodsukkeret gjennom testperioden, undersøkte gruppen igjen skjermtiden logget av testpersonens telefon av MVP-en.

#### **3.9.2 Effektmål**

For å undersøke måloppnåelsen for effektmålene beskrevet i forprosjektsplanen, ble det blant annet gjennomført to brukertester for å undersøke om MVP-en sitt design var intuitiv og lett å ta i bruk. Disse brukertestene er i vedlegg F. For å oppnå målet om at sluttbruker skal ønske å bruke applikasjonen videre, har testpersonen besvart dette ved slutten av testperioden.

For å måle effektmålene om at det skulle være lettere og mindre tidkrevende å regulere blodsukkeret, samt at diabetikeren skulle bli bedre regulert, ble det først undersøkt hvor lang tid testpersonen brukte på å regulere blodsukkeret i ukene før og i slutten av testperioden, mens MVP-en var i bruk. Deretter brukte gruppen data om variasjonen i blodsukkeret til testpersonen, samt et estimat på HbA1c-verdier før MVP-en ble tatt i bruk. Variasjonen og HbA1c ble deretter sammenlignet med verdien på slutten av prosessen, og gruppen ser på utviklingen av HbA1c-verdien før og under bruk av MVP-en, for å måle og estimere dens godhet.

---

## 3.10 Systemutvikling

### 3.10.1 Utvikling

Som nevnt i delseksjon 3.2.3, måtte gruppen utvikle et rammeverk, MVP-en, for å kunne samle inn den nødvendige dataen. Det var et fokus for gruppen å utvikle et godt design for å minske tiden som blir brukt i MVP-en. Frontend-rammeverket gruppen valgte å ta i bruk er svært egnet til dette formålet, men gruppen valgte å prioritere en iOS applikasjon grunnet mangel på tid, og at fokuset ikke handlet om å gjøre MVP-en tilgjengelig for andre enn testpersonen, som har en iPhone. Verktøyet «Storyboard» i Xcode ble brukt for å skissere brukergrensesnittet, slik at gruppemedlemmene hadde en felles forståelse av hvordan MVP-en skulle se ut og operere, og hva slags komponenter som skulle implementeres. Fokuset til gruppen var ikke å lage en skalerbar løsning. MVP-en skulle holde til formålet. Det vil si å samle inn data under testperioden, på noen måneder, på en kjapp og effektiv måte. Gruppen anså det som usannsynlig at mengden data som ble generert og håndtert kom til å skape et ytelses problem. Det ble derfor ikke lagt noen tanke inn i mer passende datastrukturer, eller raskere algoritmer for å håndtere dataene, da dette ikke kom til å få noe utslag på MVP-en under testperioden. Ettersom MVP-en var nødvendig for datainnsamlingen, og datainnsamlingen var en tidskritisk prosess, ble det ikke lagt vekt på at MVP-en skulle være skalerbar. Til tross for dette, er MVP-en utviklet med en planlagt arkitektur og generelle kodeprinsipper, som å lav coupling og høy cohesjon.

### 3.10.2 Metodikk

Det ble tatt i bruk flere ulike agile metoder for å utvikle MVP-en. Gruppen konkluderte tidlig med at det var nødvendig med agile metoder for utviklingsprosessen. Dette fordi dette åpner for fleksibilitet og eventuelle endringer i prosessen, uten at det koster for mye av tid og ressurser. Det ble også diskutert om det var behov for en systemutviklingsmetode, og blant disse vurderte gruppen Scrum og Kanban. Gruppen valgte å ikke ha en rigid systemutviklingsmetode, da gruppens interne kommunikasjon og oversikt over prosess gjennom Gantt-diagrammet, var nok til å holde en god oversikt, gi den nødvendige fleksibiliteten og den gode kommunikasjonen mellom medlemmene.

Scrum ble valgt bort da gruppen så på Scrum sine seremonier, roller og aktiviteter som unødvendig og tidskrevende, og at det ville ført til unødvendig arbeid. I tillegg hadde ikke gruppen nok medlemmer til å oppfylle alle rollene som kreves i Scrum, i tillegg til at oppgavegiver er en del av gruppen. Å ha et mellomledd, i form av en «Scrum master», mellom gruppen og oppdragsgiver ble derfor sett på som svært unødvendig. Kanban ble valgt bort da gruppen ikke så et behov for blant annet å ha et Kanban board, da, som nevnt tidligere i delseksjon 3.5.1, gruppen benyttet seg av den agile metoden parprogrammering, og hadde derfor en god oversikt over utviklingen av MVP-en. Samtidig hadde gruppen en god forståelse av hva slags funksjonalitet som måtte implementeres gjennom Gantt-diagrammet, wireframe-en og visjonsdokumentet, samt diskusjon under parprogrammeringen. I tillegg var det ikke et behov for noen av gruppens medlemmer å visualisere progresjonen med et Kanban board. Det ble imidlertid brukt et issue-board på github, for å holde oversikt over mindre oppgaver. En annen grunn til at gruppen valgte å ikke benytte seg av

---

en spesifikk systemutviklingsmetodikk var fordi oppdragsgiver ikke er ekstern, men en del av gruppen, som nevnt tidligere. Dette førte til at det ikke var behov for å oppdatere/ha møter med oppdragsgiver kontinuerlig gjennom prosessen, da oppdragsgiver var en del av utviklingen og parprogrammeringen. Gruppen tok derimot inspirasjon fra Scrum og Kanban, og benyttet korte, daglige oppdateringsmøter, i de tilfellene hvor gruppen hadde adskilte oppgaver, samt for å ha en oppsummering av dagen og hvilke mål som ble oppnådd. Dette var for å holde en god oversikt over progresjonen til prosjektet, ha en oversikt over hvilke oppgaver som gjensto, og om det var nødvendig at begge arbeidet på en oppgave dersom denne tok lenger tid enn forventet. Parprogrammeringen er også en agil metode hentet fra systemutviklingsmetodikken Extreme Programming.

Gruppen tok også i bruk user stories, domenemodell, og en wireframe. Dette var for å etablere en felles forståelse for funksjonaliteten til MVP-en, hvordan denne funksjonaliteten skulle fungere, og hvordan designet av MVP-en skulle operere og se ut. Disse er beskrevet i kravdokumentasjonen, se vedlegg D.

Metoden av å hente inn ulike agile metoder på tvers av systemutviklingsmetodikker fungerte godt for gruppen. Det var god kommunikasjon grunnet parprogrammering, og daglige oppdateringsmøter, samt at begge gruppens medlemmer alltid hadde god oversikt over prosjektets fremgang grunnet Gantt-diagrammet, issue-board, parprogrammeringen og de korte daglige oppdateringsmøtene.

---

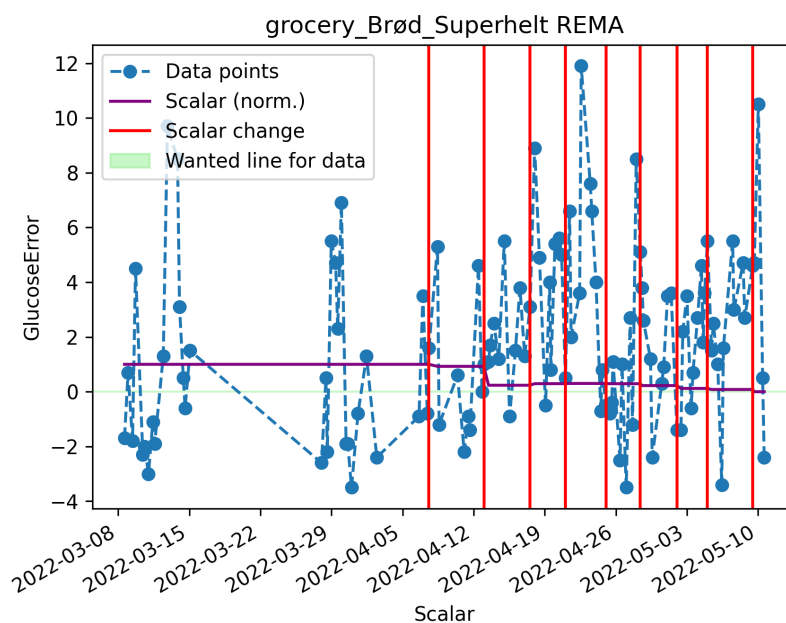
## 4 Resultater

### 4.1 Vitenskapelige resultater

#### 4.1.1 Algoritmen

Algoritmen har gjennomført flere iterasjoner på den innsamlede dataen i prosjektet, for å besvare den gitte problemstillingen. Resultatene fra algoritmen på ulike matvarer, dagsprofiler og korrigeringsinsulin vil bli redegjort for, og presentert gjennom ulike grafer. Det vil også bli presentert data fra LibreLink. Denne dataen er av testbrukers blodsukker før og under testperioden. Det vil bli undersøkt effekten MVP-en og algoritmen hadde på reguleringen av blodsukkeret til testpersonen, sett i sammenheng med reguleringen før testperioden. I mange av grafene er det både en vanlig enkel lineær regresjon og ITS analyse. Dette er gjort for å vise eventuelt hvor mye mer ITS forklarer enn den enkle lineære regresjonen. Dette kan tyde at det har vært en tydelig endring ved antatt tidspunkt som ITS modellen passer bedre til. I tilfeller hvor de forklarer omtrent like mye er de også ganske like også. Det vil da ikke ha noe å si hvilken av dem vi tar utgangspunkt i for å anta fremtidige trender. Det tas derfor alltid utgangspunkt i ITS analysen. Godheten til modellene er gitt ved RMSE-verdien deres, se delseksjon 2.2.12, som står i alle grafene det er gjort slike analyser.

##### 4.1.1.1 Matvarer

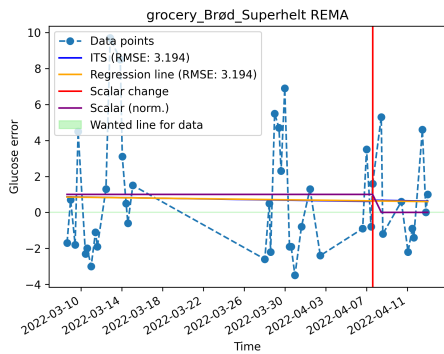


Figur 1: Total data på brød

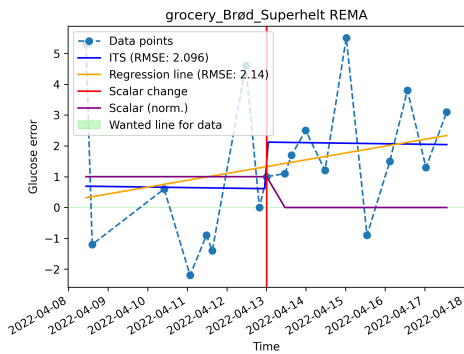
Den totale innsamlede dataen for matvaren «brød» kan observeres i figur 1. Det ble gjennomført ni iterasjoner av algoritmen på denne matvaren. I perioden fra 15.03.2022 til og med 26.03.2022 og 02.04.2022 til og med 06.04.2022 gikk data

tapt, se delseksjon 3.3.1.1. Datapunktene i grafen er glukosefeilen (glucose error) til påminnelser som brød har vært involvert i. Den lilla linjen, «Scalar (norm.)», representerer om skalaren til brød endret seg opp eller ned, og linjen er blitt normalisert (mellom 0 og 1) for å tydeliggjøre endringen. De røde vertikale linjene, «Scalar change», representerer når skalaren for brød endret seg, altså når algoritmen har gjort en endring. Den grønne linjen, kalt «Wanted line for data», er den ønskede linjen for datapunktene.

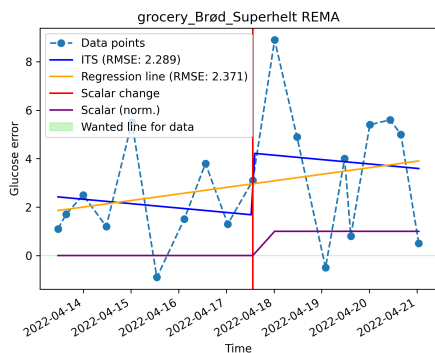
### Matvare: Brød



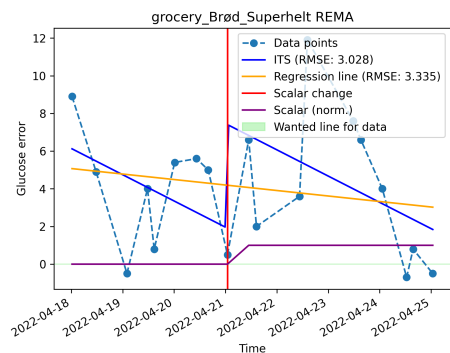
(a) Første iterasjon av algoritmen



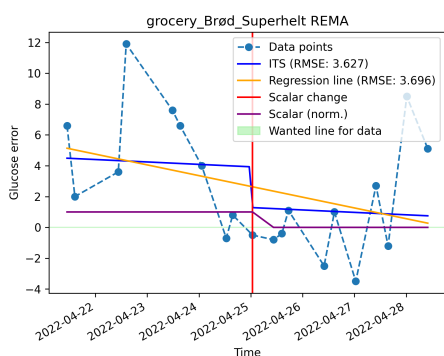
(b) Andre iterasjon av algoritmen



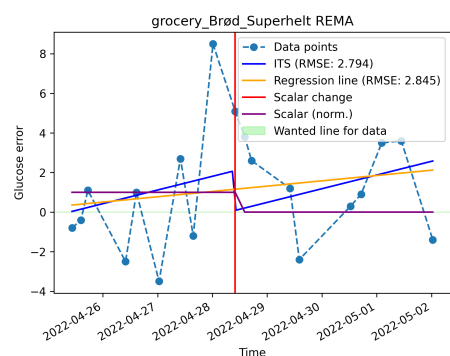
(c) Tredje iterasjon av algoritmen



(d) Fjerde iterasjon av algoritmen

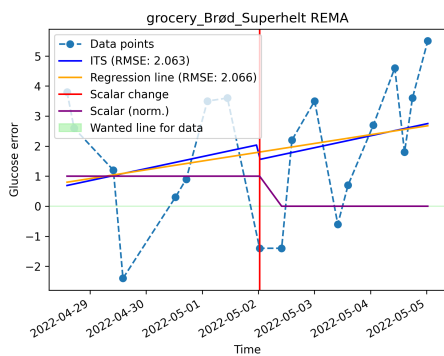


(e) Femte iterasjon av algoritmen

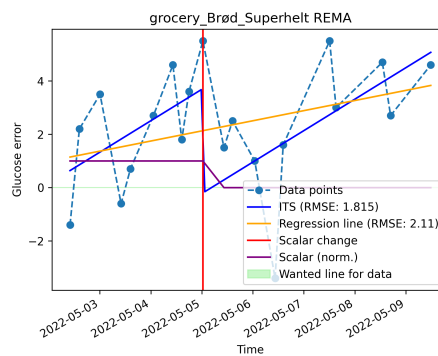


(f) Sjette iterasjon av algoritmen

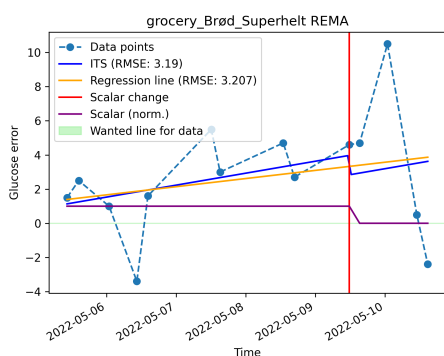
Figur 2: Utviklingen av matvaren «brød» (fortsetter på neste side)



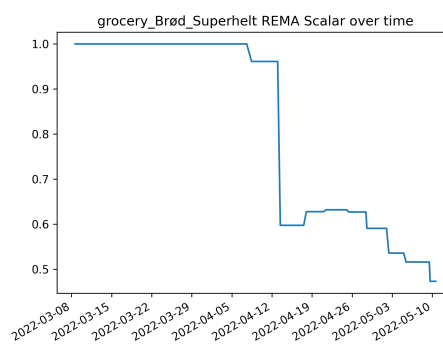
(g) Syvende iterasjon av algoritmen



(h) Åttende iterasjon av algoritmen



(i) Niende iterasjon av algoritmen



(j) Scalar over tid

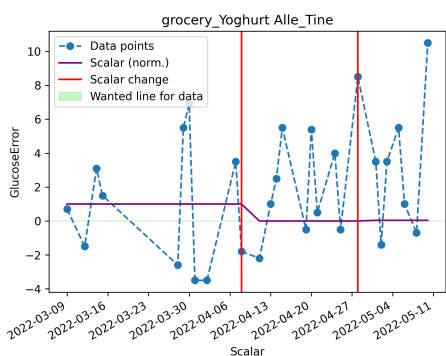
Figur 2: Utviklingen av matvaren «brød»

I figur 2 kan man nærmere observere de ulike iterasjonene av algoritmen, samt skalaren til brød over tid. I underfigurer 2a, 2b og 2g kan man observere at algoritmen justerte skalaren for brød ned, og at glukose feilen går opp. Dette er forventet oppførsel. Man kan i midlertid ikke se en stor endring i glukosefeilen etter den første iterasjonen av algoritmen vist i underfigur 2a, og selv om den har en liten stigning, holder den seg svært stabilt på samme nivå rundt 1,5. I underfigur 2b kan man observere at glukosefeilen stiger kraftig, og at den deretter har en svært slak stigning nedover. I syvende iterasjon av algoritmen, vist i underfigur 2g, kan man observere at glukosefeilen stiger litt, før den deretter har en slak stigning nedover slik som i 2b. I underfigurer 2c og 2d kan man observere at skalaren blir justert opp, og at glukose feilen også stiger, mens i underfigurer 2e og 2f iterasjon, skjer det motsatte og begge blir justert ned. Den tredje iterasjonen av algoritmen, vist i underfigur 2c, ser man at glukosefeilen stiger kraftig, før den deretter avtar slakt, mens i underfigur 2d stiger glukosefeilen ennå mer i starten før den har en bratt nedstigning. I underfigur 2e kan man observere at glukosefeilen blir raskt redusert, før den har en svært slak nedstigning. I sjetende iterasjon av algoritmen, vist i underfigur 2f, ser man i midlertid det motsatte, der den først senker seg før den har en kraftigere stigning. I den åttende og niende iterasjonen, underfigurer 2h og 2i, kan man observere at skalaren justeres ned, og at glukosefeiles først stilles ned for så å stige. I den åttende iterasjonen er dette mer ekstremt enn i niende. I underfigur 2j, ser man at skalaren for brød startet med utgangspunkt i 1,0, og raskt justerte seg ned til omtrent 0,5, og det kan se ut til at den kommer til å fortsette å synke med små steg.

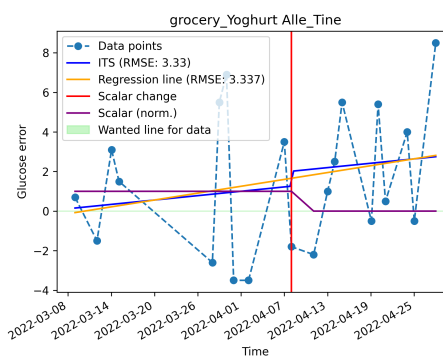
De samme grafene, med likt oppsett, har blitt generert for alle de ulike matvarene

det ble samlet inn nok datapunkter på. Det er også blitt generert like grafer for dagsprofilene.

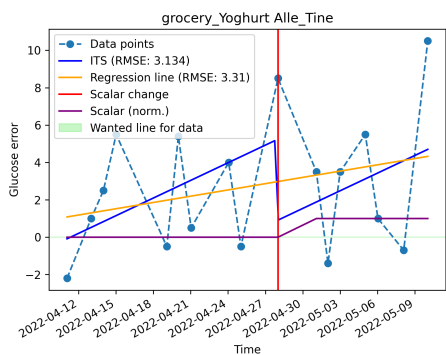
### Matvare: Yoghurt



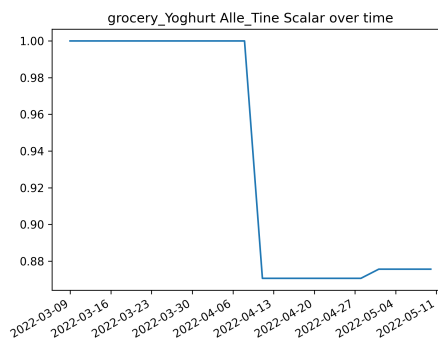
(a) Total data på yoghurt



(b) Første iterasjon av algoritmen



(c) Andre iterasjon av algoritmen

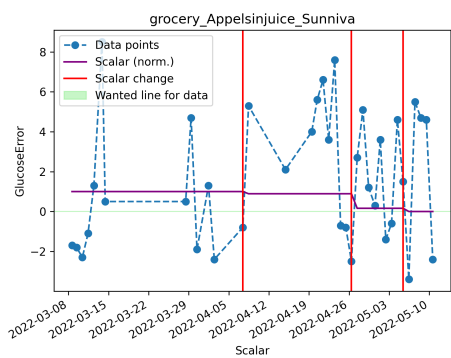


(d) Scalar over tid

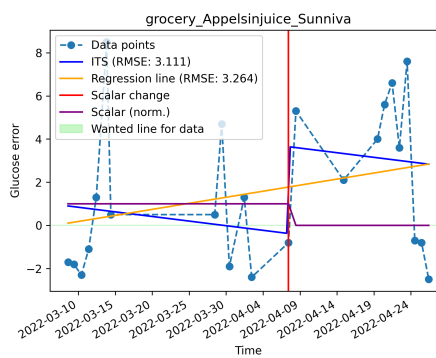
Figur 3: Utviklingen av matvaren «yoghurt»

I figur 3 kan man observere like justeringer av algoritmen, som i figur 2. Det er betraktelig mindre data på denne matvaren, yoghurt, og den har kun gjennomgått to iterasjoner av algoritmen. I underfigur 3b kan man observere at skalaren for yoghurt blir justert ned, og at glukose feilen stiger, og holder denne slakte stigende trenden. Under den andre iterasjonen av algoritmen, vist i underfigur 3c, kan man observere at det motsatte skjer, nemlig at skalaren blir justert opp, og at glukose feilen først går kraftig ned, før den igjen, slakt begynner å stige. I underfigur 3d ser man at skalaren hadde en bratt nedgang fra 1,0 til omtrent 0,86 i første iterasjon. I den andre iterasjonen steg den fra omtrent 0,86 til 0,88, en svært mye mindre endring enn i første iterasjon.

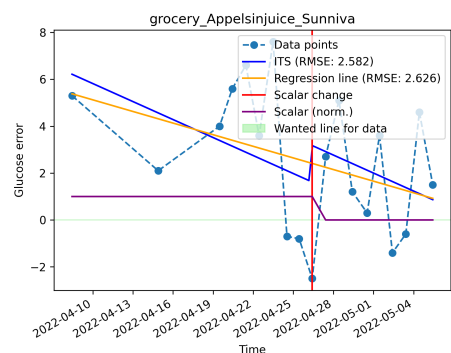
## Matvare: Appelsinjuice



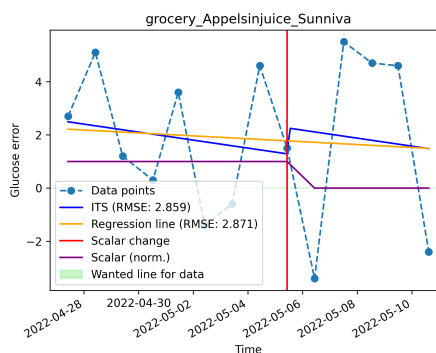
(a) Total data på appelsinjuice



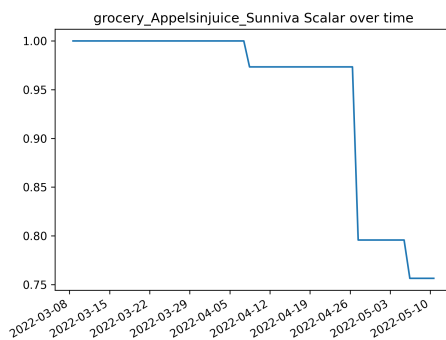
(b) Første iterasjon av algoritmen



(c) Andre iterasjon av algoritmen



(d) Tredje iterasjon av algoritmen



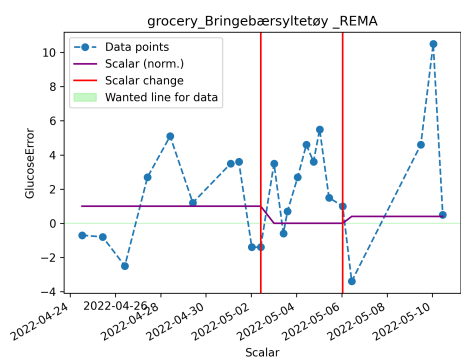
(e) Scalar over tid

Figur 4: Utviklingen av matvaren «appelsinjuice»

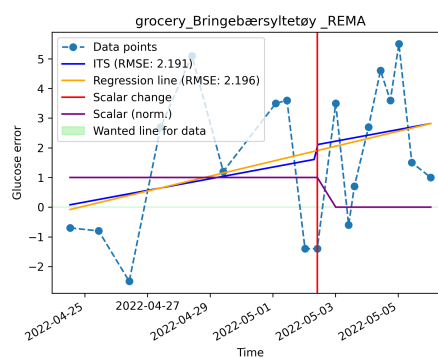
For matvaren appelsinjuice er det også blitt gjennomført kun tre iterasjoner med algoritmen. I underfigurene 4b, 4c, og 4d kan man observere at skalaren for appelsinjuice blir justert ned og at glukose feilen først stiger, før den senere får en slak nedgang. Skalaren til appelsinjuice over tid, vist i underfigur 4e, viser at skalaren ble justert ned fra 1,00 til rundt 0,75.



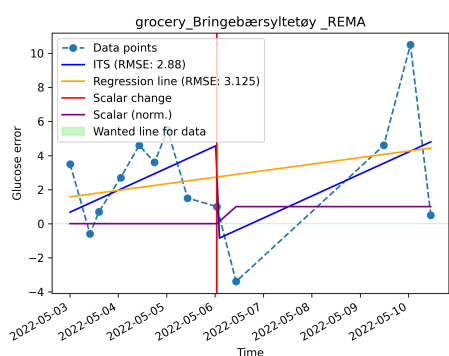
## Matvare: Bringebærsyltetøy



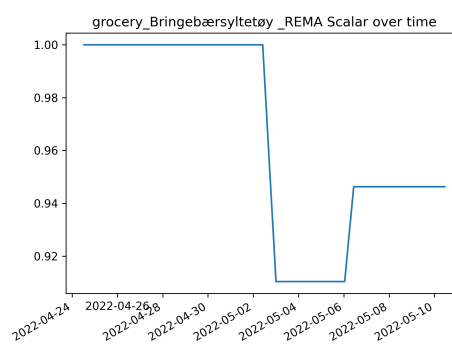
(a) Total data på bringebærsyltetøy



(b) Første iterasjon av algoritmen



(c) Andre iterasjon av algoritmen



(d) Scalar over tid

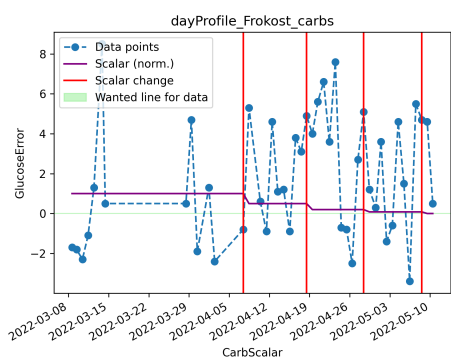
Figur 5: Utviklingen av matvaren «bringebærsyltetøy»

Bringebærsyltetøy er matvaren med minst datapunkter, og den har kun gjennomgått to iterasjon av algoritmen, slik som vist i underfigur 5a. I underfigur 5b ser man at skalaren justerte seg ned, og at glukose feilen også justerte seg ned, men at den deretter fikk en slak stigning. I andre iterasjon steg skalaren og dette medførte at glukosefeilen først sank, for deretter å stige, se underfigur 5c. I tillegg ser man at skalaren til bringebærsyltetøy justerte seg ned fra 1,00 til rundt 0,95.

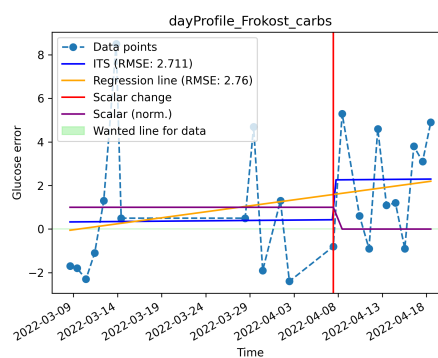
### 4.1.1.2 Dagsprofiler

Det eksisterer grafer for hver dagsprofil som ble opprettet under testperioden. Hver av disse dagsprofilene har deretter grafer for karbohydrat skalaren, og glukose skalaren, se delseksjon 3.7.2.

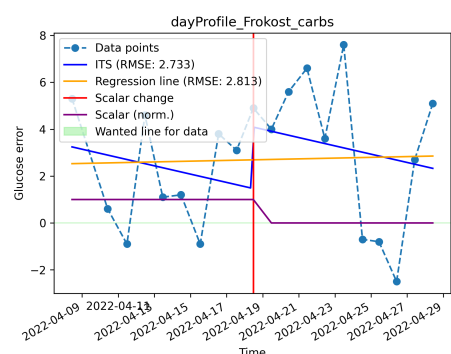
## Dagsprofil: Frokost - CarbScalar



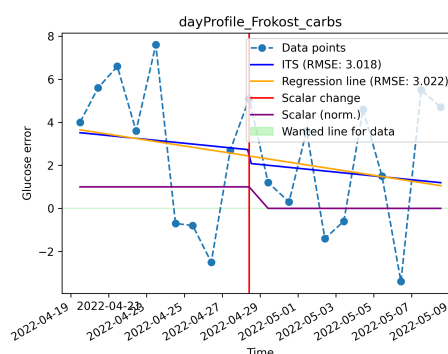
(a) Total data på frokost CarbScalar



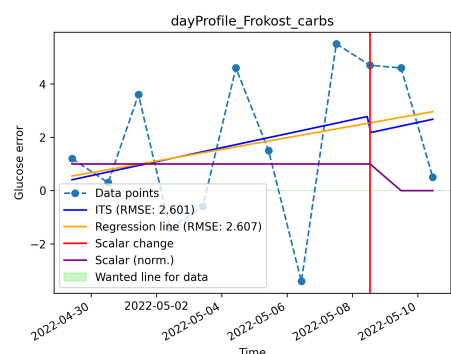
(b) Første iterasjon av algoritmen



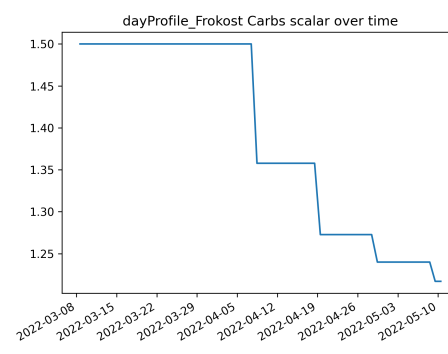
(c) Andre iterasjon av algoritmen



(d) Tredje iterasjon av algoritmen



(e) Fjerde iterasjon av algoritmen

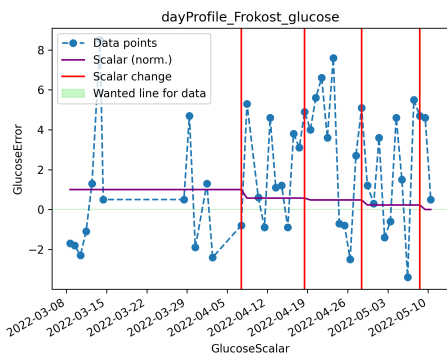


(f) Scalar over tid

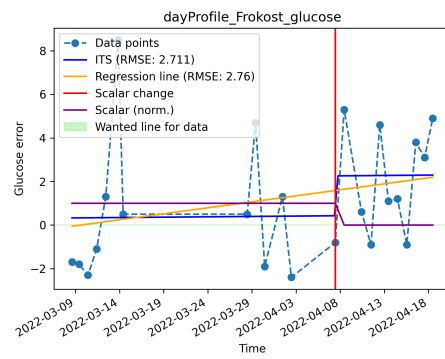
Figur 6: Utviklingen av CarbScalar ved frokost

Som man kan observere er det likt oppsett for dagsprofil grafene som for matvarer. Dagsprofilen «frokost» har gjennomgått fire iterasjoner av algoritmen, som vist i underfigur 6a. I underfigur 6b kan man se at skalaren ble justert opp, og at glukosefeilen også økte, for deretter å holde seg stabilt på det nivået. Underfigur 6c og 6d viser at skalaren for karbohydrater ble justert ned, og at glukosefeilen først økte, for deretter å ha en slak avtaking. I den fjerde iterasjonen, kan man observere det samme, bortsett fra at feilen økte i stedet for å synke, se underfigur 6e. Skalaren over tid, justerte seg fra 1,00 til omtrent 1,20, som observert i underfigur 6f.

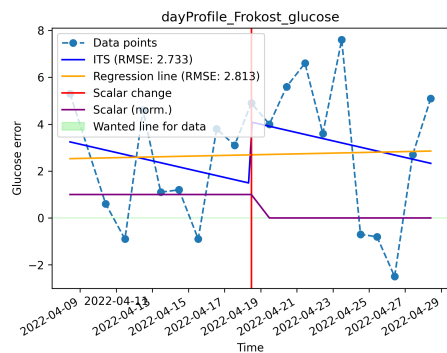
## Dagsprofil: Frokost - GlucoseScalar



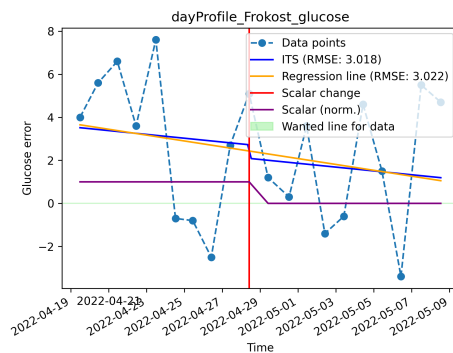
(a) Total data på frokost GlucoseScalar



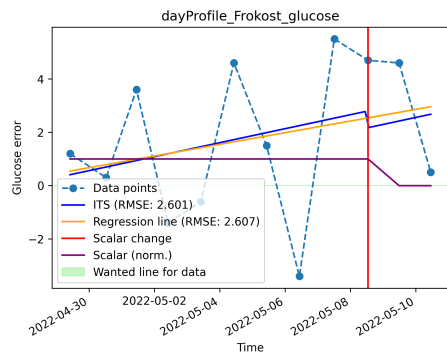
(b) Første iterasjon av algoritmen



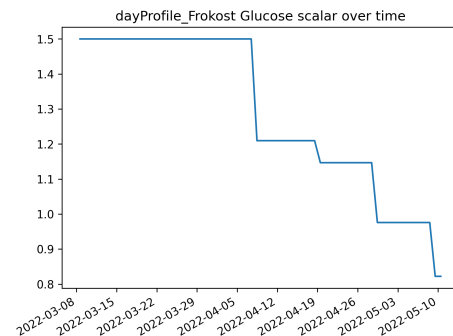
(c) Andre iterasjon av algoritmen



(d) Tredje iterasjon av algoritmen



(e) Fjerde iterasjon av algoritmen

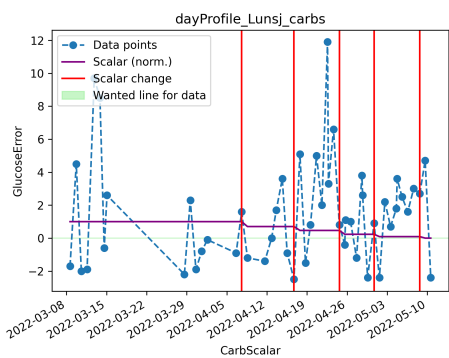


(f) Scalar over tid

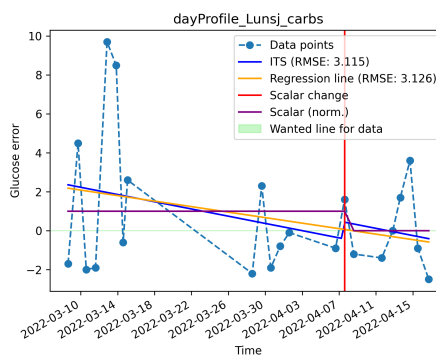
Figur 7: Utviklingen av GlucoseScalar ved frokost

I likhet med figur 6, har figur 7 også gjennomgått fire iterasjoner med algoritmen. I den første iterasjonen kan man observere i underfigur 7b at skalaren ble justert ned, og at glukosefeilen økte, og stabiliserte seg. Nesten det samme skjer i underfigurer 7c og 7d, bortsett fra at glukosefeilen øker, for deretter å synke relativt kraftig. Skalaren over tid viser at skalaren gradvis har sunket fra 1,5 til rett under 1,0.

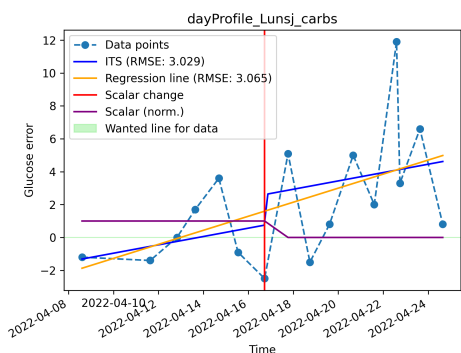
## Dagsprofil: Lunsj - CarbScalar



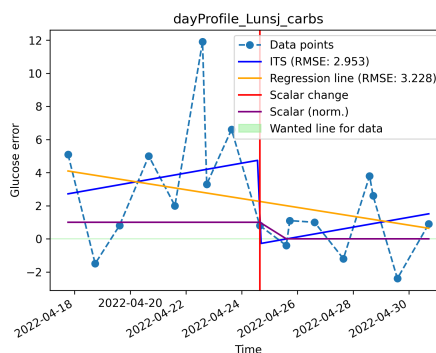
(a) Total data på lunsj CarbScalar



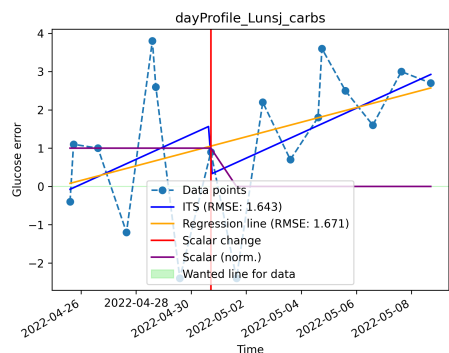
(b) Første iterasjon av algoritmen



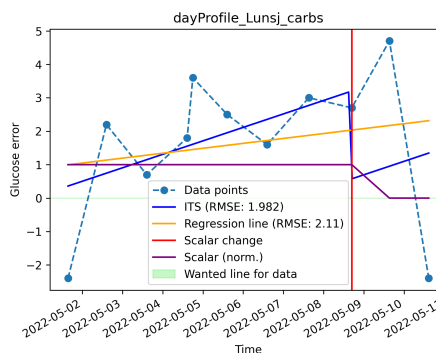
(c) Andre iterasjon av algoritmen



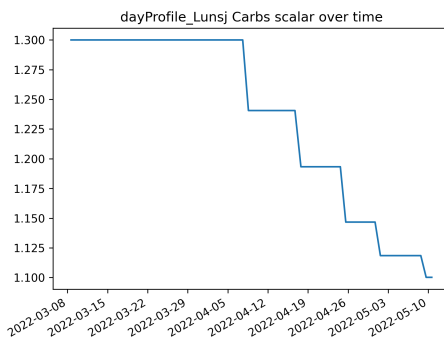
(d) Tredje iterasjon av algoritmen



(e) Fjerde iterasjon av algoritmen



(f) Femte iterasjon av algoritmen

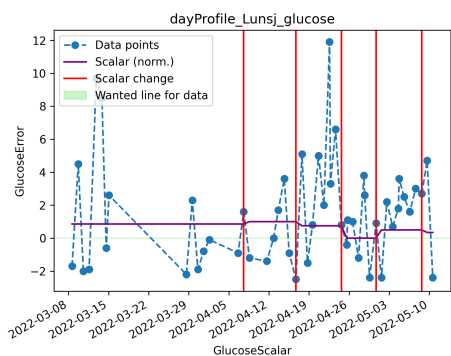


(g) Scalar over tid

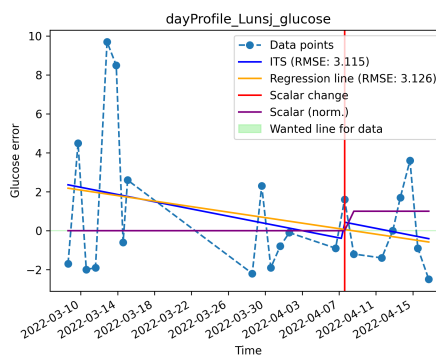
Figur 8: Utviklingen av CarbScalar ved lunsj

Dagsprofilen lunsj har hatt fem iterasjoner av algoritmen. I underfigur 8b ser man at både skalaren justeres opp og at glukosefeilen øker. I neste iterasjon, som observert i underfigur 8c, justeres skalaren ned og glukosefeilen øker. I underfigurer 8d, 8e og 8f kan man observere at skalaren justeres ned, og at glukosefeilen først senkes, før den deretter stiger. Skalaren over tid har sunket fra 1,30 til rundt 1,00.

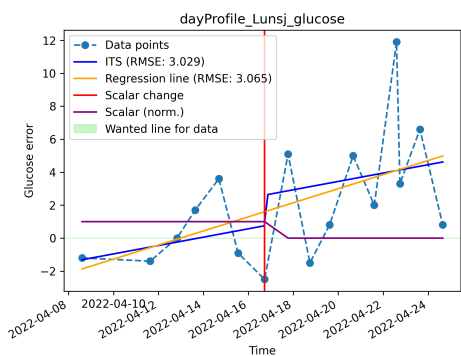
### Dagsprofil: Lunsj - GlucoseScalar



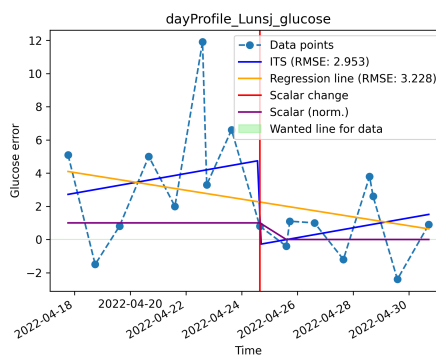
(a) Total data på lunsj GlucoseScalar



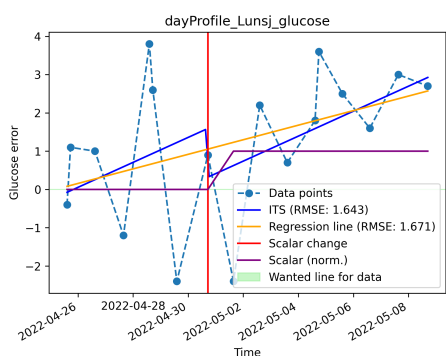
(b) Første iterasjon av algoritmen



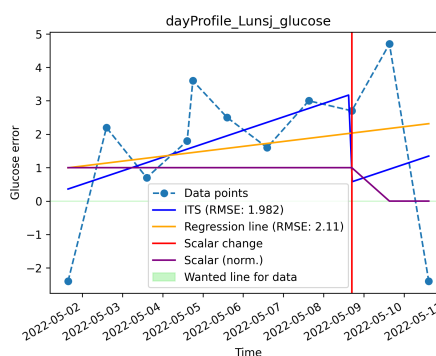
(c) Andre iterasjon av algoritmen



(d) Tredje iterasjon av algoritmen

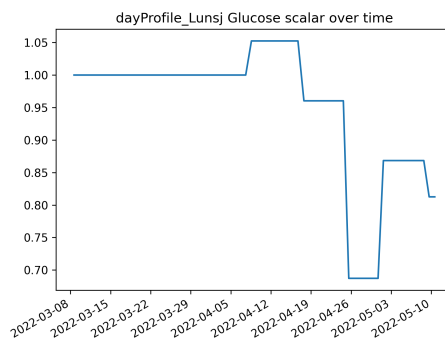


(e) Fjerde iterasjon av algoritmen



(f) Femte iterasjon av algoritmen

Figur 9: Utviklingen av GlucoseScalar ved lunsj (fortsetter på neste side)

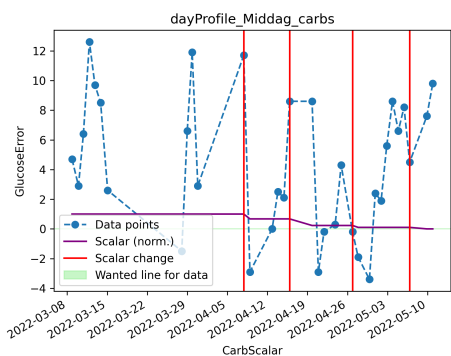


(g) Scalar over tid

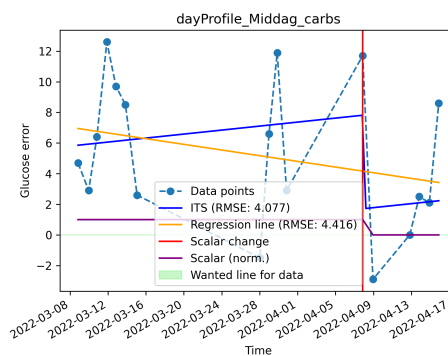
Figur 9: Utviklingen av GlucoseScalar ved lunsj

Algoritmen har også gjennomført fem endringer for glukose skalaren for lunsj. I underfigur 9b kan man se at skalaren øker i verdi, og at glukosefeilen først stiger, for deretter å avta. I andre iterasjon, underfigur 9c, synker skalarverdien og glukosefeilen øker. I tredje og femte iterasjon synker igjen skalarverdien, men denne gangen synker også glukosefeilen, for deretter at den slakt stiger igjen. Dette blir vist i underfigurer 9d og 9f. I den fjerde iterasjonen, vist i underfigur 9e, øker skalarverdien, og glukosefeilen senkes først, før den deretter stiger. Underfiguren for skalaren over tid, viser hvordan den har justert seg fra 1,00 til et sted mellom 0,85 og 0,80.

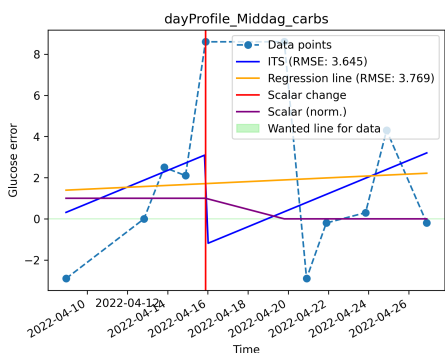
## Dagsprofil: Middag - CarbScalar



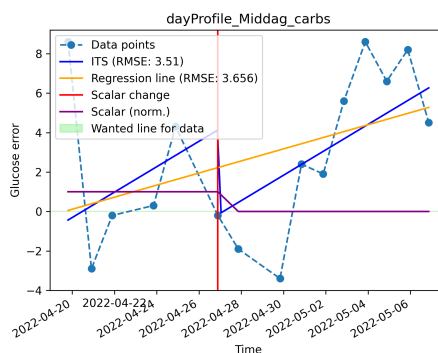
(a) Total data på middag CarbScalar



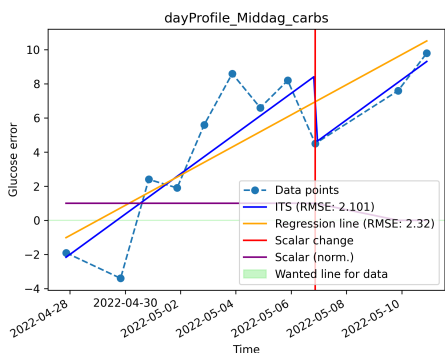
(b) Første iterasjon av algoritmen



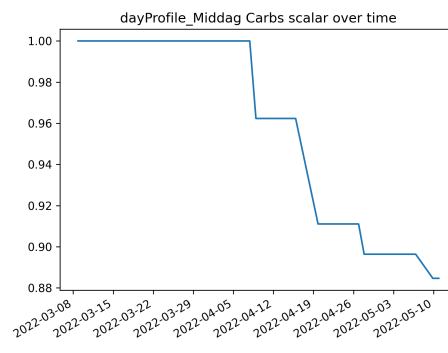
(c) Andre iterasjon av algoritmen



(d) Tredje iterasjon av algoritmen



(e) Fjerde iterasjon av algoritmen

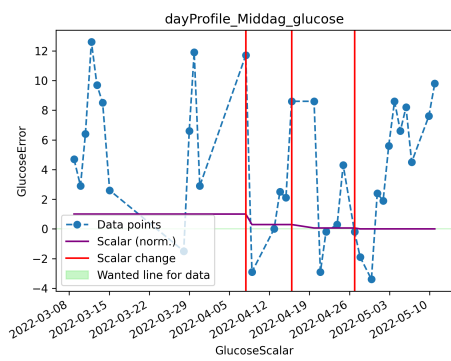


(f) Scalar over tid

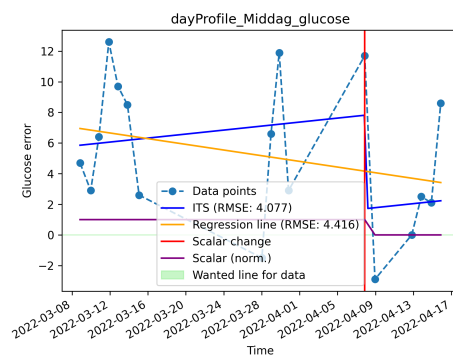
Figur 10: Utviklingen av CarbScalar ved middag

Middag har gjennomgått fire iterasjoner av algoritmen. I alle iterasjonene har skalaren blitt justert ned, samt at glukosefeilen også har sunket, for deretter å stige oppover. I underfigur 10f kan man observere at skalaren har endret seg fra 1,00 til rett under 0,90.

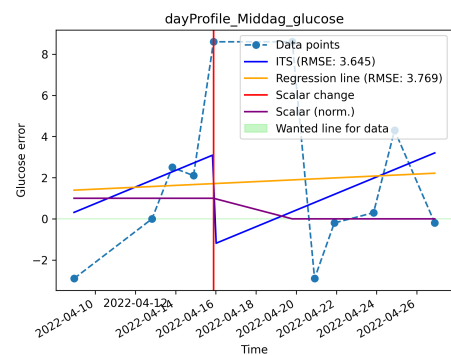
## Dagsprofil: Middag - GlucoseScalar



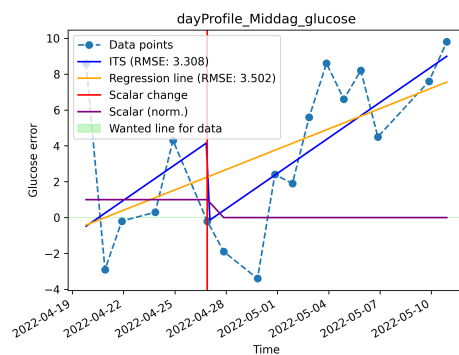
(a) Total data på middag glukoseskalar



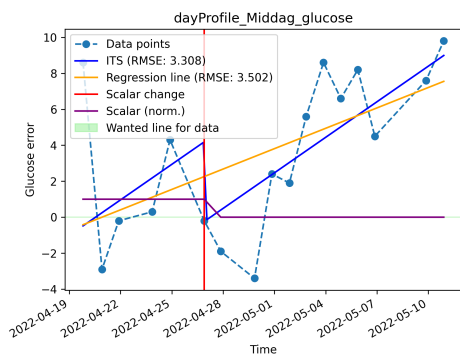
(b) Første iterasjon av algoritmen



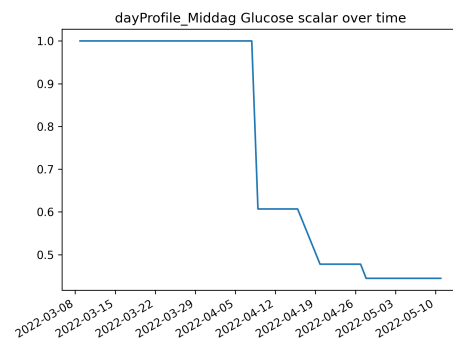
(c) Andre iterasjon av algoritmen



(d) Tredje iterasjon av algoritmen



(e) Fjerde iterasjon av algoritmen



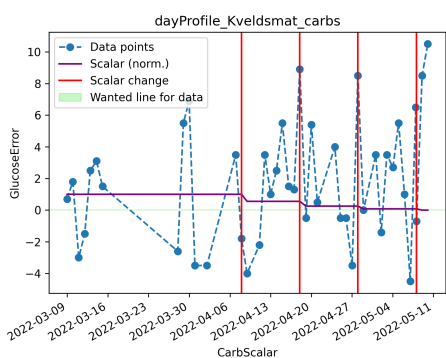
(f) Scalar over tid

Figur 11: Utviklingen av GlucoseScalar ved middag

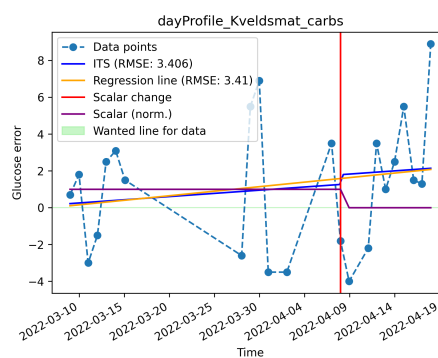
Glukoseskalaren for dagsprofilen «middag» har også gjennomgått fire iterasjoner med algoritmen, og man kan observere store likheter med figur 10. I alle tre iterasjonene, som vist i underfigurer 11b, 11c og 11d, kan man observere at skalaren justeres ned, og at glukosefeilen først senkes, for deretter å øke.



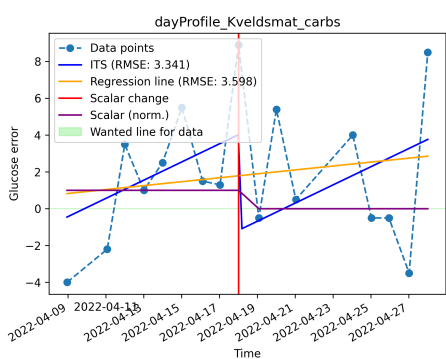
## Dagsprofil: Kveldsmat - CarbScalar



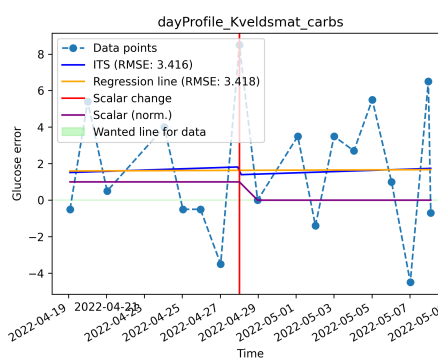
(a) Total data på kveldsmat karbohydrat-skalar



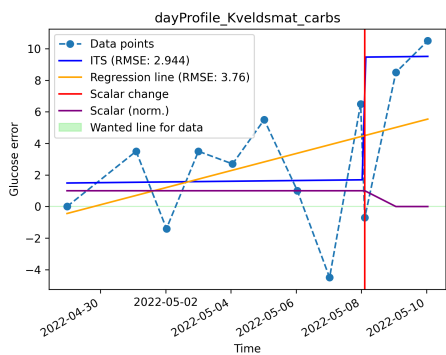
(b) Første iterasjon av algoritmen



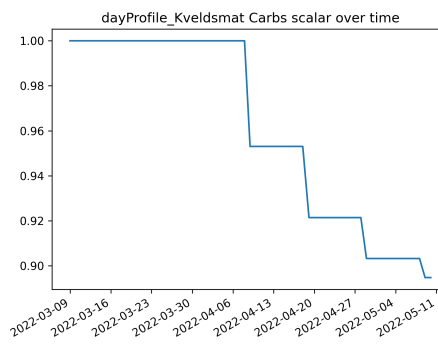
(c) Andre iterasjon av algoritmen



(d) Tredje iterasjon av algoritmen



(e) Fjerde iterasjon av algoritmen

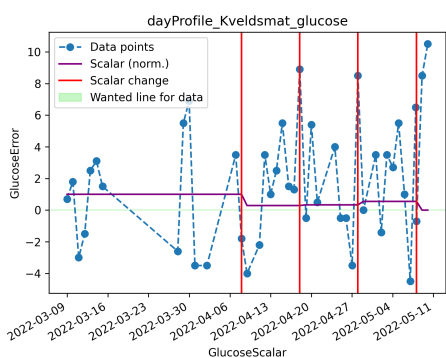


(f) Scalar over tid

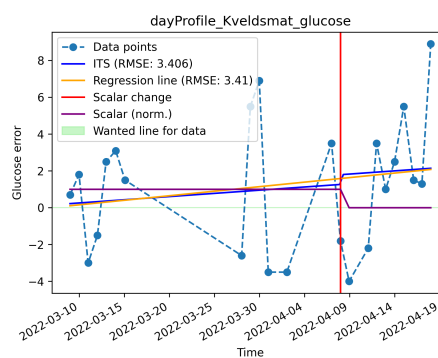
Figur 12: Utviklingen av CarbScalar ved kveldsmat

Dagsprofilen «kveldsmat» har gjennomgått fire iterasjoner av algoritmen. I alle iterasjonene har skalaren blitt justert ned. I den første og fjerde iterasjonen, underfigurer 12b og 12e, økte imidlertid glukosefeilen, mens i andre og tredje, vist i underfigurer 12c og 12d, iterasjon sank denne for deretter å stige. Underfigur 12f viser hvordan skalaren har sunket fra 1,00 til rett under 0,90.

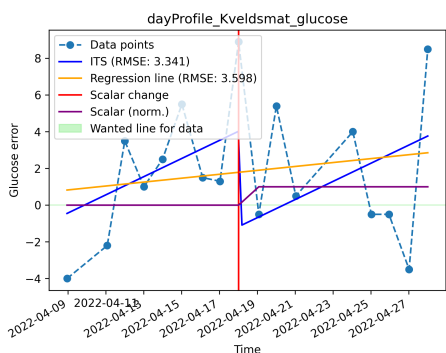
## Dagsprofil: Kveldsmat - GlucoseScalar



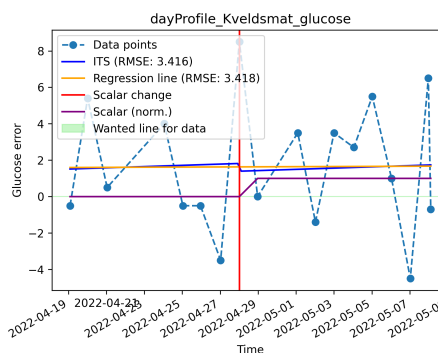
(a) Total data på kveldsmat glukoseskalar



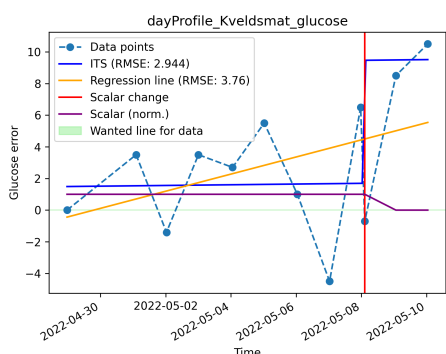
(b) Første iterasjon av algoritmen



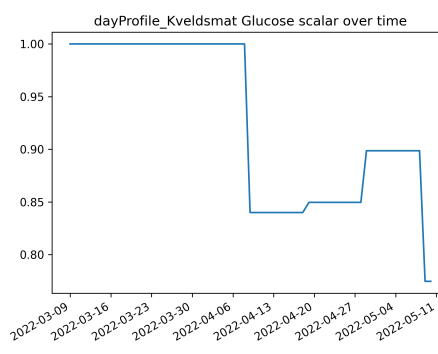
(c) Andre iterasjon av algoritmen



(d) Tredje iterasjon av algoritmen



(e) Fjerde iterasjon av algoritmen

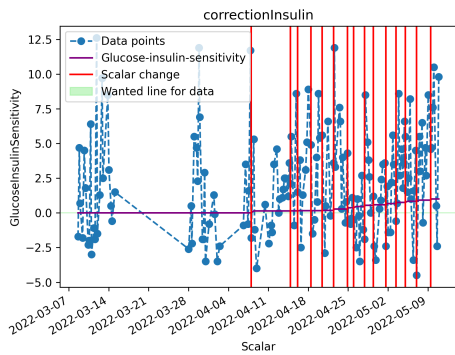


(f) Scalar over tid

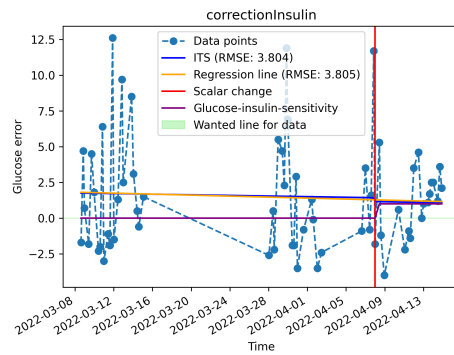
Figur 13: Utviklingen av GlucoseScalar ved kveldsmat

Som tidligere har glukoseskalaren gjennomført like mange iterasjoner som karbohydratskalaren for også denne dagsprofilen. Her ser man skalaren har justert seg ned i første og fjerde iterasjon. I tillegg, som vist i 13b og 13e, øker også glukosefeilen. Underfigurer 13c og 13d viser at skalaren justeres opp, og at glukosefeilen i begge synker før de øker i verdi. Skalarens endring vises i underfigur 13f, der den faller fra 1,00 til under 0,80.

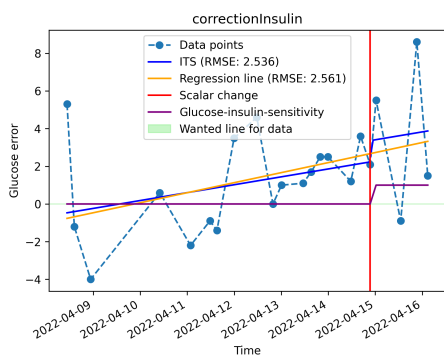
### 4.1.1.3 Korrigeringsinsulin



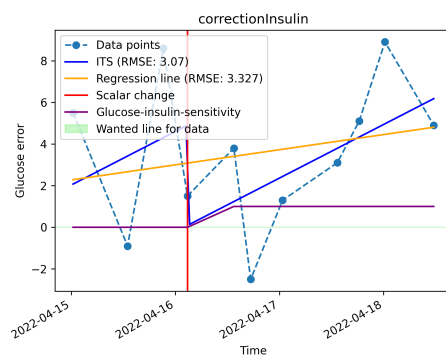
(a) Total data



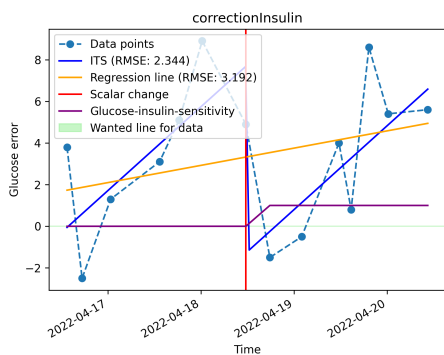
(b) Første iterasjon av algoritmen



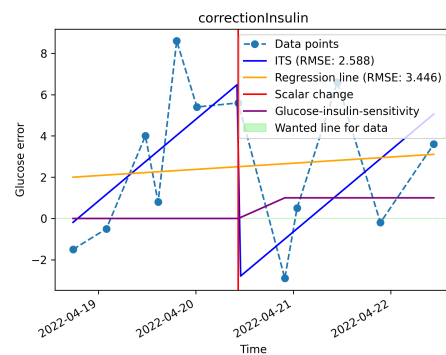
(c) Andre iterasjon av algoritmen



(d) Tredje iterasjon av algoritmen

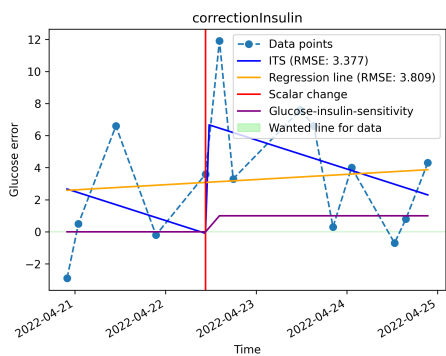


(e) Fjerde iterasjon av algoritmen

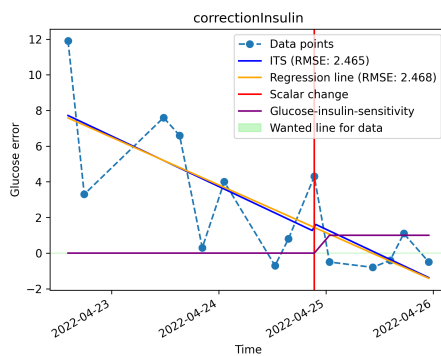


(f) Femte iterasjon av algoritmen

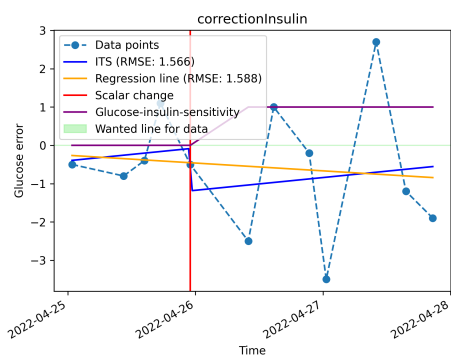
Figur 14: Utviklingen av korrigeringsinsulin (fortsetter på neste side)



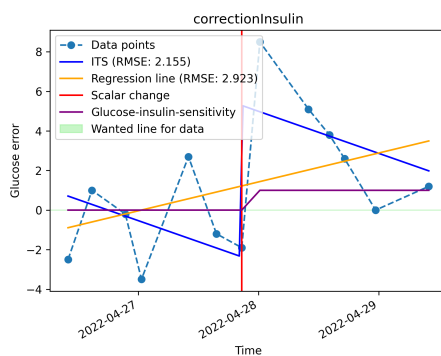
(g) Sjette iterasjon av algoritmen



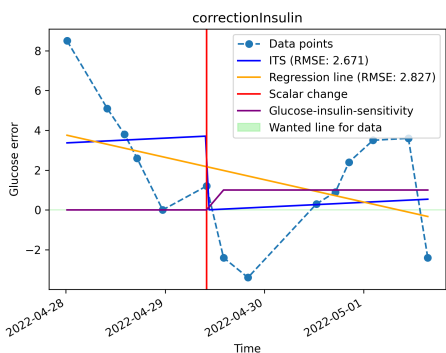
(h) Syvende iterasjon av algoritmen



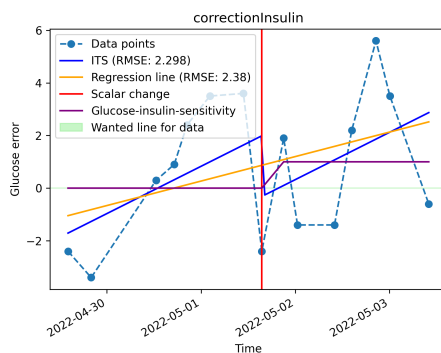
(i) Åttende iterasjon av algoritmen



(j) Niende iterasjon av algoritmen

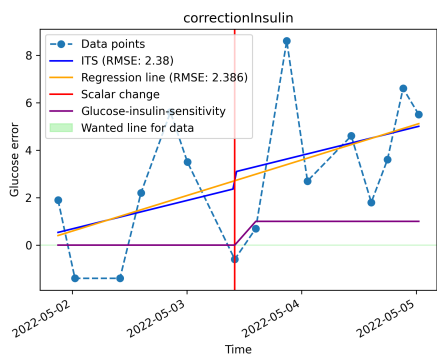


(k) Tiende iterasjon av algoritmen

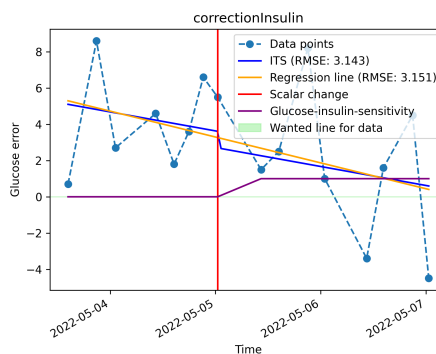


(l) Ellefte iterasjon av algoritmen

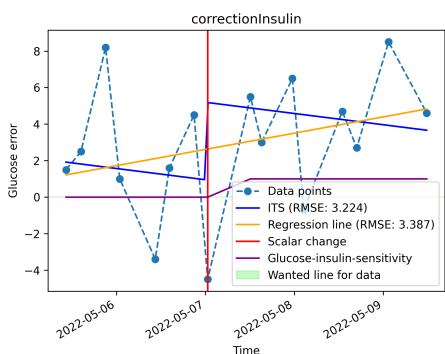
Figur 14: Utviklingen av korrigeringsinsulin (fortsetter på neste side)



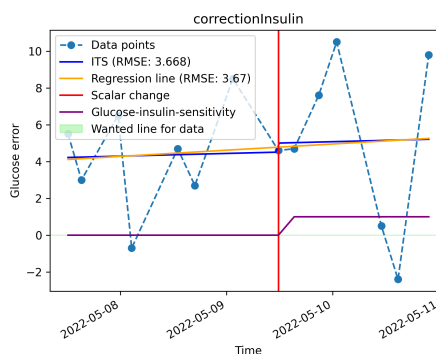
(m) Tolvte iterasjon av algoritmen



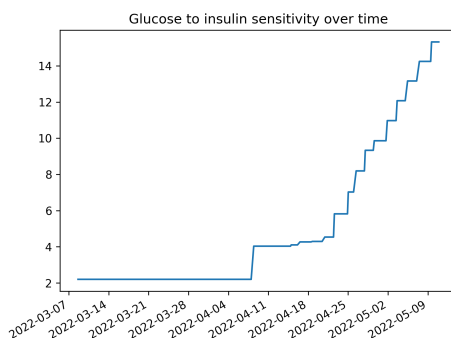
(n) Trettende iterasjon av algoritmen



(o) 14. iterasjon av algoritmen



(p) 15. iterasjon av algoritmen



(q) Sensitivitet mellom glukose og insulin

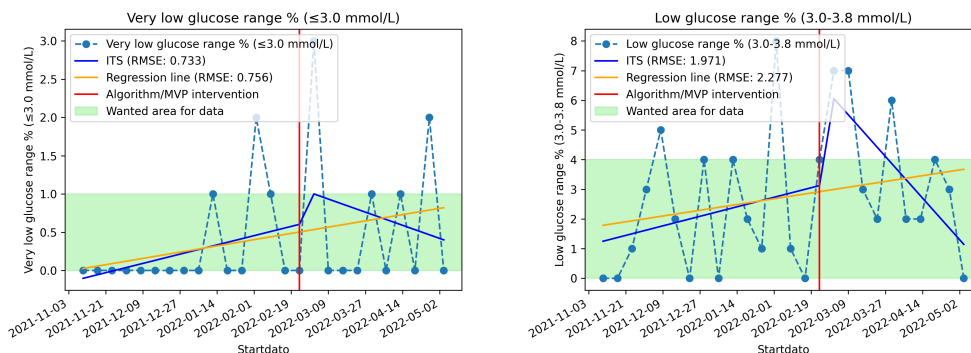
Figur 14: Utviklingen av korrigeringsinsulin

Figur 14 viser oversikten og utviklingen av korrigeringsinsulin, og i underfigur 14a kan man observere at korrigeringsinsulin gjennomgikk femten iterasjoner med algoritmen. I figur 14 kan man observere at skalaren konsekvent har justert seg opp. I første og trettende iterasjon, 14b og 14n, kan man observere at glukosefeilen hadde en liten nedgang. I andre, tolvte og femtende iterasjon steg glukosefeilen, se underfigurer 14c, 14m og 14p. I underfigurer 14d, 14e, 14f, 14i, 14k og 14l kan man se at glukosefeilen først justerte seg ned, før den ofte hadde en kraftig stigning opp. I den sjettede, syvende, niende og fjortende iterasjonen øker glukosefeilen, for deretter å ha en nedgang, se underfigurer 14g, 14h, 14j, og 14o. Underfigur 14q viser hvordan sensitiviteten mellom glukose og insulin har endret seg over tid. Det starter med en stor endring ved introduksjonen av algoritmen for så at det er slakt fram til 22. april hvor den plutselig begynner å stige kraftig. Dette er verdien som først ble regnet ut

med 100-regelen beskrevet i delseksjon 2.1.

#### 4.1.1.4 LibreLink

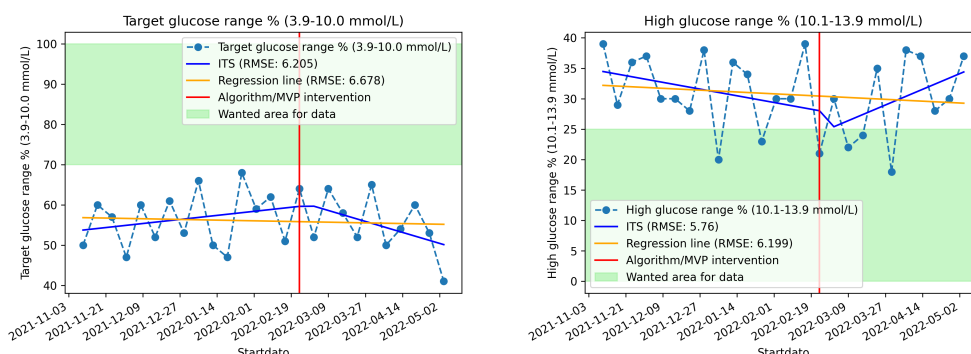
Fra sensoren til testpersonen var det mulig å hente ut ulik data som HbA1c-verdi, variasjon i blodsukker og hvor lenge testpersonen var innenfor ulike soner (intervaller i blodsukkeret) gjennom perioden.



(a) Veldig lave glukoseverdier

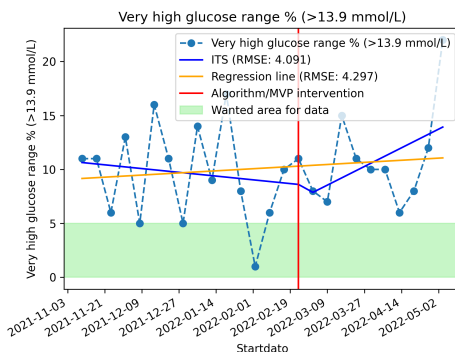
(b) Lave glukoseverdier

Figur 15: Oversikt over hvor lenge testpersonen oppbevarte seg i de ulike glukosesonene (forsetter på neste side)



(c) Ønsket glukoseverdier

(d) Høye glukoseverdier

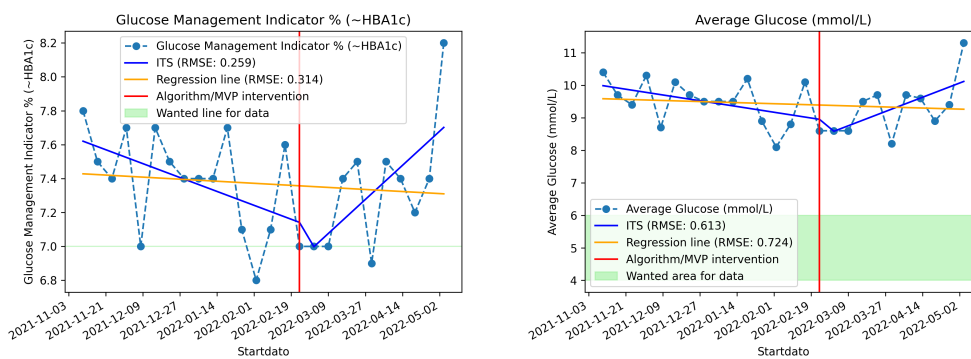


(e) Veldig høye glukoseverdier

Figur 15: Oversikt over hvor lenge testpersonen oppbevarte seg i de ulike glukosesonene

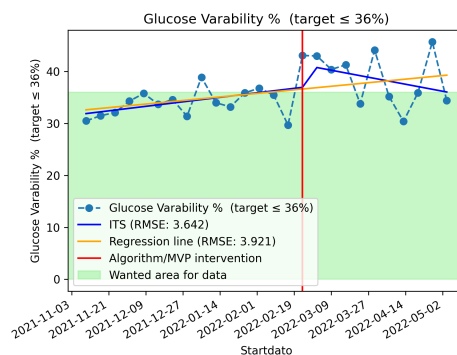
I figur 15 kan man observere det samme oppsettet som for figurene for matvarer og dagsprofiler. I underfigur 15c kan man se de blå datapunktene, «Target glucose range % (3.9-10.0 mmol/l)», som representerer prosenten av hvor ofte testpersonen befant seg i denne sonen. Videre kan man observere den røde vertikale linjen, «Algorithm/MVP intervention», som indikerer introduksjonen av MVP-en. Det grønne området, kalt «Wanted area for data», representerer hvor mange prosent av tiden man ønsker å være innenfor denne sonen. Den oransje linjen, «Regression line (RMSE: X.XXX)», er regresjonslinjen for dataen. Den blå linjen, «ITS (RMSE: Y.YYY)», er ITS linjen for datapunktene.

I figur 15 kan man observere tiden, i prosent, testpersonen var innenfor de fem ulike sonene: veldig lavt blodsukker, lavt blodsukker, ønsket blodsukker, høyt blodsukker og veldig høytblodsukker. Som man kan observere i underfigur 15a, øker først antallet med tilfeller innen sonen med veldig lavt blodsukker. Deretter kan man observere en slak nedgang. Man kan se at denne sonen har likheter med lavt blodsukker, vist i underfigur 15b. Også her kan man se en økning i hvor lenge testpersonen befinner seg i sonen, før den har en bratt nedgang. I underfigur 15c kan man observere at etter introduksjon av algoritmen, synker tiden innenfor denne sonen, og den fortsetter å ha en slak nedgang. For sonen med høye glukoseverdier, underfigur 15d, kan man observere en nedgang ved MVP-en sin introduksjon, før den begynte å øke. I underfigur 15e, som representerer de veldig høye glukoseverdiene, kan man observere at denne hadde en økning.



(a) Utviklingen av HbA1c-verdi

(b) Utvikling av gjennomsnittlig glukoseverdi



(c) Utvikling av variasjonen i blodsukkeret

Figur 16: Glukosehåndtering

Figur 16 viser hvor godt blodsukkeret har blitt håndtert, både før og etter MVP-en

sin introduksjon. Underfigur 16a viser hvordan HbA1c-verdien til testpersonen har utviklet seg fra rundt 10.11.2021 til siste innhentet datapunkt 10.05.2022. Ved introduksjonen av MVP-en kan man observere en liten nedgang, før verdien begynte å øke. Gjennomsnittlig blodsukker blir vist i underfigur 16b, og her kan man observere det samme som i underfigur 16a. Underfiguren for variasjon i blodsukkeret, 16c, viser at det var en økning ved MVP-en sin introduksjon, men at den senere har hatt en nedgang.

#### 4.1.1.5 Korrelasjonsmatriser

	AG	GMI	GV	VH	H	T	L	VL
AG	1	0,99	-0,56	0,72	0,63	-0,75	-0,72	-0,45
GM		1	-0,54	0,74	0,61	-0,75	-0,71	-0,44
GV			1	0,05	-0,72	0,40	0,54	0,47
VH				1	0,01	-0,45	-0,49	-0,34
H					1	-0,83	-0,31	-0,06
T						1	0,19	-0,04
L							1	0,54
VL								1

Tabell 1: AG - Average Glucose (mmol/l), GMI - Glucose Management Indicator % (HbA1c), GV - Glucose Variability % (target  $\leq 36\%$ ), VH - Very high glucose range % ( $>13.9$  mmol/L), H - High glucose range % (10.1-13.9 mmol/L), T - Target glucose range % (3.9-10.0 mmol/L), L - Low glucose range % (3.0-3.8 mmol/L), VL - Very low glucose range % ( $\leq 3.0$  mmol/L). Basert på før og under testperioden.

I tabell 1 kan man observere en korrelasjonsmatrise for all data i LibreLink, både før og etter MVP-ens introduksjon. Man kan for eksempel observere en korrelasjon mellom sonen for ønskede blodsukkerverdier og sonen for høye blodsukkerverdier er -0,83.

	AG	GMI	GV	VH	H	T	L	VL
AG	1	0,99	-0,64	0,73	0,53	-0,70	-0,60	-0,57
GM		1	-0,64	0,73	0,52	-0,69	-0,60	-0,55
GV			1	-0,02	-0,82	0,62	0,36	0,18
VH				1	-0,11	-0,31	-0,53	-0,58
H					1	-0,85	0,02	0,02
T						1	-0,08	0,06
L							1	0,61
VL								1

Tabell 2: AG - Average Glucose (mmol/l), GMI - Glucose Management Indicator % (HbA1c), GV - Glucose Variability % (target  $\leq 36\%$ ), VH - Very high glucose range % ( $>13.9$  mmol/L), H - High glucose range % (10.1-13.9 mmol/L), T - Target glucose range % (3.9-10.0 mmol/L), L - Low glucose range % (3.0-3.8 mmol/L), VL - Very low glucose range % ( $\leq 3.0$  mmol/L). Basert på før testperioden.



Tabell 2 viser korrelasjonen for all data i Librelink før algoritmen og MVP-en ble introdusert.

	<b>AG</b>	<b>GMI</b>	<b>GV</b>	<b>VH</b>	<b>H</b>	<b>T</b>	<b>L</b>	<b>VL</b>
<b>AG</b>	1	0,99	-0,55	0,77	0,73	-0,87	-0,90	-0,37
<b>GM</b>		1	-0,52	0,80	0,71	-0,86	-0,91	-0,38
<b>GV</b>			1	-0,01	-0,71	0,41	0,63	0,53
<b>VH</b>				1	0,20	-0,62	-0,60	-0,27
<b>H</b>					1	-0,85	-0,75	-0,07
<b>T</b>						1	0,73	-0,05
<b>L</b>							1	0,43
<b>VL</b>								1

Tabell 3: AG - Average Glucose (mmol/l), GMI - Glucose Management Indicator % (HbA1c), GV - Glucose Variability % (target  $\leq 36\%$ ), VH - Very high glucose range % ( $>13.9$  mmol/L), H - High glucose range % (10.1-13.9 mmol/L), T - Target glucose range % (3.9-10.0 mmol/L), L - Low glucose range % (3.0-3.8 mmol/L), VL - Very low glucose range % ( $\leq 3.0$  mmol/L). Basert på under testperioden.

I tabell 3 kan man observere korrelasjonen for all data i Librelink under bruk av MVP-en og algoritmen. Man kan observere at under MVP-ens introduksjon, har lave verdier og ønskede verdier en korrelasjon på 0,73, i forhold til før MVP-en. Da eksisterte det en korrelasjon på -0,08. Det er også en betraktelig større sammenheng mellom både LG og AG, og LG og GM, under MVP-en enn før dens introduksjon.

## 4.2 Ingeniørfaglige resultater

I begynnelsen av prosjektet ble det utarbeidet en forprosjektsplan som beskriver blant annet resultat- og effektmålene for oppgaven. Ingen av resultatmålene ble nådd, men tre av fire effektmål ble oppnådd. I tillegg ble det skrevet et visjonsdokument med funksjonelle og ikke-funksjonelle krav til MVP-en. De fleste av disse kravene er oppfylt, men enkelte ble forkastet, grunnet tid og at andre områder, som ble sett på som viktigere, trengte mer fokus.

### 4.2.1 Resultatmål

Av resultatmålene beskrevet i forprosjektsplanen ble ingen av målene oppnådd. Disse målene var da:

- Sluttbruker skal bruke 50% mindre tid på regulering av blodsukker enn de gjorde før.
- Sluttbrukers HbA1c-verdi skal gå ned med 0.2 prosentpoeng.

Testbrukeren brukte omtrent 40% mindre tid på reguleringen av blodsukkeret i uken, se delseksjon 4.2.1.1. Ved prosjektets slutt var testbrukerens HbA1c-verdi gått opp 0,3 prosentpoeng, mens underveis før 22. april var den uendret, se delseksjon 4.2.1.2.

---

#### 4.2.1.1 Tidsbruk

Testpersonen anslo ved prosjektets start at han brukte omtrent tre timer i uken på å regulere og følge opp blodsukkeret, basert på skjermtiden logget av telefonen av applikasjonene «Kalkulator» og «LibreLink», sensor applikasjonen, som loggfører blodsukkeret. Dette blir presentert i tabell 4. Tiden testpersonen har brukt på å finne ut hvor mange karbohydrater en matvare har, er ikke blitt tatt med i tabellen, da dette som regel er neglisjerbart, i tillegg til at gruppen ikke hadde en god måte å måle denne tiden på.

Uke (kalender)	Kalkulator	LibreLink	Totalt
3	100 min	80 min	180 min
4	102 min	81 min	183 min
5	101 min	79 min	180 min
6	97 min	84 min	181 min

Tabell 4: Tabell over tidsbruk i ulike uker før MVP-en

Uke (kalender)	MVP-en	LibreLink	Totalt
15	51 min	54 min	105 min
16	50 min	52 min	102 min
17	53 min	52 min	105 min
18	60 min	62 min	122 min
19	48 min	62 min	110 min

Tabell 5: Tabell over tidsbruk i ulike uker med MVP-en

Etter introduksjon av MVP-en brukte testpersonen gjennomsnittlig 108,8 minutter i uken, som er omtrent 1,8 timer i uken. Dette ble målt med skjermtiden logget av telefonen til testpersonen på MVP-en, vist i tabell 5. Dette er en reduksjon på omtrent 40% i uken.

#### 4.2.1.2 HbA1c

Da testpersonen begynte å ta i bruk MVP-en den 8. mars, var testpersonens HbA1c-verdi på 7,2 prosent, samt at variasjon i blodsukkeret var 38,3 prosent, basert på de fire forekommende ukene. Utviklingen av HbA1c-verdien fra januar og gjennom perioden vises i figur 16a. Ved slutten av testperioden var HbA1c-verdien på 7,5 prosent, samt at variasjonen i blodsukkeret var 38,1 prosent de siste fire ukene. Før 22. april, hvor en akselerert endring i «insulin-til-glukose»-forholdet forekom, var HbA1c-verdien 7,2 prosent med en variasjon på 36,8 prosent de siste fire ukene.

---

### 4.2.2 Effektmål

Av effektmålene beskrevet i forprosjektsplanen ble tre av disse fire målene nådd. Målene beskrevet var:

- Lettere og mindre tidkrevende å regulere blodsukkeret for diabetikere.
- Diabetikere skal bli bedre regulert.
- Designet av applikasjonen skal være enkelt og intuitivt.
- Sluttbruker ønsker å bruke applikasjonen videre.

Det første målet ble nådd ved at sluttbruker brukte mindre tid på å regulere blodsukkeret, slik som beskrevet i delseksjon 4.2.1. I tillegg ble det bekreftet av testpersonen at ved å ta i bruk MVP-en, ble det lettere å håndtere blodsukkeret. Testpersonens opplevelse av MVP-en var at den gjorde at det var lettere og mindre tidkrevende å regulere blodsukkeret.

Det andre målet ble ikke nådd, da både variasjonen i blodsukkeret og HbA1c-verdien økte. Variasjonen i blodsukkeret sank fra 38,3 til 38,1 prosent og HbA1c-verdien steg fra 7,2 til 7,5 prosent, som vises i figurene 16a og 16c. Selv ikke med de tidligere resultatene fra før 22. april ble dette kravet oppfylt da variasjonen gikk fra 38,3 til 36,8 prosent og HbA1c-verdien var uendret.

Målet om at designet av applikasjonen skulle være enkelt og intuitivt ble testet gjennom brukertester som beskrevet i delseksjon 3.8. Resultatene fra disse brukertestene var at brukerne opplevde at MVP-en var oversiktlig, intuitiv og lett å ta i bruk. Referater fra brukertestene er i vedlegg F.

Det siste målet ble oppnådd da testpersonen gjennom prosessen uttrykte at vedkommende ville fortsette å bruke MVP-en ved prosjektets slutt. Det kom også frem i den ene brukertesten, se vedlegg F.2, at denne brukeren hadde et ønske om å ta i bruk applikasjonen selv.

### 4.2.3 Frontend

Det ble lagt vekt på at applikasjonen skulle minimere tiden en diabetiker bruker på å regulere blodsukkeret. For å oppnå dette målet, ble det vektlagt å lage et godt og oversiktlig brukergrensesnitt, uten at det skulle ta for mye tid å utvikle slik at det gikk utover algoritmen. Målet om å redusere tiden brukt på å regulere blodsukkeret med 50% ble ikke nådd, men tiden ble fortsatt redusert med omtrent 40%. Blant kravene til brukergrensesnittet ble det lagt vekt på å kunne registrere satt insulin, administrere matvarer og dagsprofiler, en kalkulator for beregningen av insulin, logging av satt insulin, og påminnelser. I tillegg, med lav prioritet, skulle man kunne søke etter en gitt matvare. MVP-en har funksjonalitet for å opprette, endre og slette både matvarer og dagsprofiler, samt søke etter en gitt matvare. I tillegg eksisterer det funksjonalitet for å beregne insulin, og registreringen av det insulinet som ble satt, samt logging av insulinmengden. Funksjonalitet for å opprette påminnelser tre timer etter satt insulin, er også blitt implementert. Man kan legge matvarer i en liste, og skrive inn nåværende blodsukker for å kalkulere nødvendig insulinmengde. Matvarer og dagsprofiler har sine egne tabber, der bruker har god oversikt over hva de har

---

registrert av både matvarer og dagsprofiler. Alle loggene testpersonen har blir også vist i et kalender format, der man må bla mellom ulike dager for å finne en ønsket logg. Disse målene blir derfor sett på som nådd. Det ble også laget en egen tabb, ment som et hjelpeverktøy for utviklerne under debugging. Her vises tiden brukt i MVP-en, samt «karbohydrat-til-insulin»- og «glukose-til-insulin»-forholdet. Den viser også en «Correction only scalar», men denne har ikke blitt brukt i testperioden, da testpersonen alltid har lagt til matvarer ved logging.

Noen av målene med lavere prioritetsgrad som ble forkastet var events, tidsprofiler og oppkobling av MVP-en mot en sensor og/eller en insulinpumpe.

#### **4.2.4 Ikke-funksjonelle egenskaper og andre krav**

Kravet om at rammeverket skulle utvikles med Xamarin Forms og C# ble oppfylt, og gruppen tok utgangspunkt i iPhone 11, da det var denne mobile enheten testpersonen hadde. Gruppen fant imidlertid en annen løsning for å laste opp MVP-en til en mobil enhet. I stedet for å bruke TestFlight, tok gruppen i bruk AdHoc som gjorde at MVP-en kunne lastes opp på den mobile enheten rett fra utviklingsmiljøet, Visual Studio for Mac.

Et annet krav omhandlet logging av systemet og unntakshåndtering rundt verdier som påvirker eller er knyttet til den estimerte insulinmengden. Dette kravet anses som oppnådd da man kan se unntakshåndteringen i kildekoden. Det ble ikke tatt i bruk tradisjonell logging, men debug-logging ble anvendt på enkelte steder. Kravet om å ta i bruk en relasjonsdatabase er også oppnådd. Dette kan sees ved databasemodellen gruppen produserte, og teknologien SQLite som ble tatt i bruk i rammeverket, er en relasjonsdatabase. Databasemodellen vises i systemdokumentasjonen i vedlegg D.

#### **4.2.5 Kravdokumentasjon**

I kravdokumentasjonen blir det spesifisert at man skal kunne endre matvarer, logger og dagsprofiler. Det er implementert funksjonalitet for dette, men gruppen har bevist valgt å ikke ta det i bruk. Dette blir beskrevet mer i detalj i delseksjon 5.1.

#### **4.2.6 Brukertester**

Det ble gjennomført to semi-strukturerte brukertester, se vedlegg F. Fra brukertestene fikk gruppen undersøkt og målt hvor brukervennlig, intuitivt og lett MVP-en var å bruke. Den ene brukeren har diabetes, mens den andre ikke har. Til tross for dette, mente begge brukerne at designet og funksjonaliteten var lett å ta i bruk, samt at begge klarte å bruke all funksjonaliteten slik gruppen hadde ment for at den skulle brukes. Brukeren med diabetes hadde mer forståelse for hva og hvorfor funksjonaliteten var som den var, og hvorfor den var der, enn den brukeren uten diabetes gjorde. Likevel opplevde begge at det som skjedde når de brukte ulik funksjonalitet, var det de forventet skulle skje. De mente også at designet var pent og oversiktlig. De hadde litt ulike førsteinntrykk av MVP-en, der brukeren uten diabetes først fikk inntrykk av at det var en bank applikasjon, da fargene minnet om DNB sin mobilapplikasjon. Den andre brukeren antok at MVP-en var diabetes relatert, og at hun kunne regne ut insulin for karbohydrater.

---

Begge brukerne trodde at «Day profile/Dagsprofil»-tabben ville vise hva de hadde spist i løpet av dagen, hva slags blodsukkerverdier de har hatt i løpet av dagen, og hvordan enhetene har blitt estimert.

MVP-en er et vektøy begge brukerne anså om fornuftig, og de ville anbefalt den til andre. Brukeren med diabetes uttrykte også at dette var et verktøy hun selv hadde lyst til å bruke, og følte seg spent etter brukertesten, da hun så frem til at andre kunne ta MVP-en i bruk.

## 4.3 Administrative resultater

### 4.3.1 Fremdriftsplan og måloppfyllelse

Svært tidlig i prosessen ble det utarbeidet et Gantt-diagram for å holde oversikt over arbeidsoppgaver, og prosjektets progresjon. Ved starten av prosjektet var Gantt-diagrammet slik som beskrevet i vedlegg D (forprosjektplan). Det ble nødvendig med noen justeringer underveis i prosessen. Ved prosjektets slutt var Gantt-diagrammet som beskrevet i vedlegg D (prosessdokumentasjon).

Forandringene som blir dokumentert under er de større endringene gjennomført i løpet av prosjektet. Som man kan observere i det reviderte Gantt-diagrammet, har de fleste datoene for gjennomføring av ulike oppgaver blitt endret, men de har blitt gjennomført i den planlagte rekkefølgen.

En av justeringene i Gantt-diagrammet var blant annet at gruppen implementerte logger for tidligere satt insulin, før algoritmen. I tillegg ble algoritmen implementert før gruppen lagde og testet resultatanalysen.

Resultatanalysen ble ikke startet på før 27.04.22, og dette er 22 dager etter original planlagt dato. Arbeidsoppgaven «Lage og teste resultatanalysen» ble gjennomført 04.05.2022.

Gruppen begynte å skrive på hovedrapporten allerede i uke 13. Gruppen fordelte deler av arbeidet i denne perioden, for å effektivisere prosessen. For eksempel, ble arbeidet fordelt slik at en arbeidet på hovedrapporten, mens en annen skrev tester til koden.

Gruppen hadde ikke tatt hensyn til at det skulle skrives tester til koden, da Gantt-diagrammet først ble utarbeidet, og dette ble gjort i perioden 12.04.2022 til 25.04.2022, som vist i det reviderte Gantt-diagrammet. Arbeidsoppgaven heter «Lage tester».

En milepæl i prosessen var å implementere algoritmen for insulinmengde. Originalt hadde gruppen satt av tre dager til å gjennomføre dette, der siste dag var mål for milepælen, og planen var å finpusse algoritmen derfra. I virkeligheten tok algoritmen omtrent elleve dager å implementere, inkludert omarbeid av kode, samt bugfiksing. Gruppen valgte å ikke finpusse eller justere algoritmen. Dette da testperioden ikke kunne deles i flere deler, da datagrunnlaget hadde blitt for lite. Før algoritmen ble implementert ble det et behov for en omfattende omskriving («rework») av koden. Det ble introdusert DAO-klasser til alle modellene. Det ble også lagt til lag i arkitekturen: «Service» og «Repository». Dette diskuteres i delseksjon 5.7.

Originalt var det en arbeidsoppgave kalt «Modeller og sett opp database». Denne ble endret navn til «Modeller database», og implementasjonen av databasen for de

---

ulike aspektene ved koden, ble implementert underveis. For eksempel, da klassen for matvarer ble implementert i koden, ble den tilhørende databasetabellen implementert.

I tillegg ble de to arbeidsoppgavene «Implementer registrering av satt insulin» og «Implementere logg av tidligere satte insulinmengder», kombinert til en oppgave, «Administrering av logger (registrering av satt insulin og logger av tidligere satt insulinmengder)».

Det ble også opprettet en annen arbeidsoppgave «Brukertester».

Det var også flere arbeidsoppgaver i Gantt-diagrammet, som «Korrekturlesing og finpussing av hovedrapport» og «Finne relevant faglig arbeid til hovedrapport», som går innunder arbeidsoppgaven «Hovedrapport», og ble gjort i den samme perioden som denne. I det reviderte Gantt-diagrammet er disse blitt slått sammen til et punkt: «Hovedrapport». I tillegg er arbeidsoppgaven om å finne relevant faglig arbeid, kun gruppen som satte av tid til å finne relevante kilder for allerede eksisterende kunnskap, som for eksempel til statistikk og diabetes. Gruppen gjorde de tre nevnte arbeidsoppgavene om hverandre og parallelt mens hovedrapporten ble skrevet.

Gruppen oppdaget omtrent midt i prosessen, at presentasjonen av bacheloroppgaven skulle utarbeides og leveres sammen med hovedrapporten og annen dokumentasjon. Dette var heller ikke tatt med i førsteutkastet til Gantt-diagrammet.

#### **4.3.2 Planlegging og tidsbruk**

Timebruket samsvarer godt med Gantt-diagrammet. Gruppen har kontinuerlig loggført hvor lang tid en aktivitet i Gantt-diagrammet faktisk tok, i forhold til den planlagte tiden. Som man kan observere i Gantt-diagrammet, var det noe feilberegning i hvor lang tid noen oppgaver kom til å ta. Ettersom denne feilberegningen både besto av oppgaver som tok lenger tid enn planlagt, og andre oppgaver som tok kortere tid, balanserte dette seg godt ut, og gruppen fikk gjennomført alle de planlagte aktivitetene i tillegg til de som ikke var planlagt, som dukket opp underveis i prosessen, se delseksjon 4.3.1.

Fra timelistene, se vedlegg D (prosessdokumentasjon), observerer man en jevn fordeling av arbeidet, da begge gruppens medlemmer har arbeidet omtrent like mye, samt at gruppen har arbeidet på ulike områder vedrørende oppgaven, som for eksempel dokumentasjon og utvikling. I ukesrapportene har gruppen beskrevet nøyaktig hva innenfor disse områdene de har arbeidet med. Det ble planlagt å benytte seg av parprogrammering for utviklingen, og dette kommer også frem gjennom timelistene.

#### **4.3.3 Systemutvikling**

Utviklingsprosessen gruppen har valgt å bruke består av ulike agile metoder hentet fra ulike agile systemutviklingsmetoder. Disse agile metodene består blant annet av parprogrammering, som kommer frem gjennom gruppens timelister, se vedlegg D (prosessdokumentasjon). En annen agil metode, hentet fra Kanban og Scrum, er daglige oppdateringsmøter. Ettersom gruppen kun besto av to medlemmer, var det god kommunikasjon innad i gruppen, og de daglige møtene var ikke offisielle møter,

---

men heller en svært kort oppsummering av hva gruppens medlemmer hadde gjort den dagen, og hva de planla å gjøre videre. Den daglige oppdateringen ble gjennomført på slutten av arbeidsdagen, og ble kun tatt i bruk dersom medlemmene arbeidet med ulike oppgaver. Issue-board ble også tatt i bruk under utviklingsprosessen se vedlegg E. Artefakter brukt i arbeidet har vært user stories og wireframes, som ble beskrevet i kravdokumentasjonen, se vedlegg D. Disse ble brukt for å kartlegge funksjonaliteten til rammeverket. Annet enn dette benyttet gruppen seg flittig av Gantt-diagrammet og issue-board på github, og hadde ellers svært god kommunikasjon. I kravdokumentasjonen, se vedlegg D, er det vedlagt et dokument som dokumenterer oppnådde user stories, og progresjonen på disse.

---

## 5 Diskusjon

### 5.1 Dataintegritet og datagrunnlag

Det var svært viktig for prosjektet at det ble samlet inn nok data for at algoritmen skulle kunne gjennomføre endringer, og for å undersøke hvor presis algoritmen kunne bli basert på den innsamlede dataen.

Dataen samlet inn gjennom prosjektet har vært fra én testperson. Etter råd fra veileder konkluderte gruppen med at det holdt med én testperson, og at man kunne ta et valg om å videreutvikle algoritmen, basert på de preliminare resultatene. Etersom algoritmen tilpasser seg hver testperson, ville man ikke fått et større datasett med flere endringer fra algoritmen dersom det hadde vært en større testgruppe. En større testgruppe ville derimot gitt var en oversikt over hvor mange denne algoritmen eventuelt kan hjelpe. Datainnsamlingen var svært tidskritisk i prosjektet, og gruppen forsøkte å starte prosessen så tidlig som mulig. Til tross for dette var det også svært viktig at datainnsamlingen ble gjennomført korrekt, og at det ikke eksisterte bugs i koden som ville ført til korrupt data. Grunnet dette tok gruppen god tid på å utarbeide algoritmen og MVP-en for å unngå en situasjon der all innsamlet data måtte forkastes grunnet ukorrekt kode. Av denne grunn ble innsamlingen av data startet 8. mars, og på denne datoen ble kun MVP-en uten algoritmen introdusert. Algoritmen ble introdusert 8. april. Det vil si at flere av de første iterasjonene hadde et større datagrunnlag enn de neste, da det var en måned med innsamlet data.

Det er viktig å samle inn nok data før man gjør en endring [37]. Ved å introdusere algoritmen i tidlig april var gruppen klar over at det kun var noen av matvarerene som ville samle inn nok data, for å gjennomgå flere iterasjoner av algoritmen. Det hadde vært ønskelig med mer data, da algoritmen kunne gjennomført flere iterasjoner på flere matvarer, i tillegg til at den hadde hatt mer tid til å tilpasse seg testpersonen, som ville gitt et større grunnlag for å vurdere om algoritmen fungerte. En annen grunn til at gruppen ønsket en større datainnsamlingsperiode, var for å kunne justere algoritmen dersom det ble oppdaget feil, eller aspekter ved den som kunne forbedres. Ved en lang periode for datainnsamling, kunne man gjennomført to testperioder med to forskjellige versjoner av algoritmen, men dette hadde desverre ikke gruppen mulighet til. I begynnelsen av datainnsamlingen med algoritmen, var det forventet at blodsukkeret ville variere mer, og generelt mer uregulert, frem til algoritmen, gjennom flere iterasjoner, lærte seg hvor mye insulin som trengtes til for eksempel ulike matvarer. Ved flere iterasjoner, hadde algoritmen hatt mer tid og data til å ta gode valg. I tillegg til at gruppen jobbet med algoritmen, undersøkte de også hvor mye tid man kan spare i uken med MVP-en. En måte gruppen kunne ha samlet inn mer data på, var å starte tidligere og jobbet mer i start av prosjektet, men dette var ikke praktisk mulig som man ser i Gantt-diagrammet og timelistene, se vedlegg D (prosessdokumentasjon). Gruppen gjennomførte tiltak som å arbeide i helgene for å få fortgang i prosessen, med forbehold om å ikke slite seg selv ut. Til tross for tiltaket om å arbeide mer, tidlig i prosessen, for å starte datainnsamlingen tidligere, kunne gruppen fortsatt ønsket mer data.

Som nevnt tidligere var dataintegriteten svært viktig for at algoritmen skulle oppføre seg som ønsket, at den tok korrekte valg og at disse kunne analyseres med den nødvendige dataen. Gruppen implementerte nesten all funksjonaliteten listet opp i krav- og visjonsdokumentet, men ved nærmere inspeksjon innså gruppen at deler av denne



---

funksjonaliteten, dersom den tas i bruk, vil medføre korrupt data. I kravdokumentasjonen blir det spesifisert gjennom user stories, at det skal eksistere funksjonalitet for å endre matvarer, dagsprofiler, og logger, samt legge til nye logger manuelt. Gruppen så i ettertid at dette ville medføre korrupt data. Dette er fordi, dersom en matvare først blir oppgitt med 35 karbohydrater per 100 gram, og en bruker senere endrer denne verdien til 10 karbohydrater, vil ikke algoritmen ha den nødvendige dataen for å foreslå en korrekt insulinmengde. Dette gjelder også for endring av hvilke matvarer som ble spist i logger, eller endring av tid for dagsprofiler. For å forsikre dataens integritet valgte gruppen å ikke endre noen av matvarenes, dagsprofilenes, eller loggenes tallverdier. Grunnen til at funksjonaliteten for å legge til nye logger i etterkant medfører korrupt data, er fordi i tilfellet der den nye loggen overlapper med en eller flere andre logger, som allerede kan ha blitt brukt i algoritmen. Dette vil invalidere disse dataene. Man kunne lagt inn svært kompleks logikk som fjernet endringen fram til den nye loggen, og gjorde det om. Etersom hele systemet henger så tett sammen, ville dette fått en «snøballeffekt», og man måtte gjort mange endringer som ville påvirket hva bruker valgte å sette, og dette kan ikke endres, og dataene ville blitt korrupte. Gruppen fjernet i midlertid ikke denne funksjonaliteten, da medlemmene ville ha tilgang til all funksjonalitet, dersom det var nødvendig, og det ble også sett på som unødvendig bruk av tiden. Dette spesielt da testpersonen også er en av utviklerene, og lett kunne unngå å bruke de nevnte funksjonalitetene. Det er altså mulig å endre matvarer og lignende, men gruppen har ikke tatt i bruk funksjonaliteten, for å sikre dataintegriteten.

## 5.2 Algoritmen

### 5.2.1 Konstruksjon

#### 5.2.1.1 Partisjonerings metode

Da gruppen designet feil-partisjoneringsystemet hadde de matvarer som hovedfokus, og tenkte som utgangspunkt at alle de teoretiske verdiene hadde en liten feil koblet til seg, som vil si at til noen matvarer trengte man mer insulin og andre mindre, men at disse feilene ikke var veldig store. Tanken var da at om man spiste fem brødskeer og drakk et glass juice, ville feilen i brødskeiven femdobles og ville ha størst bidrag til den totale samlede feilen. Derfor ble feilen fordelt på bidrag av insulin, som er direkte knyttet til matvarens antall. Tanken var da at selv om brød hadde perfekt dosering og all feilen kom fra juicen så ville ikke dette påvirke brødskeive-analysen, da kombinasjonene av samme mengde brød og juice må forekomme mange ganger for at det skal ha en sterk negativ effekt. Dersom dette forekommer ofte, vil denne doseringen til dette måltidet bli bedre. Uansett vil juice også få en andel av skylden hver gang, som vil endre den i riktig retning. Så med nok tid og iterasjoner ville dette jevne seg ut.

Problemet som oppsto, var at det krevdes svært mange iterasjoner for at dette skulle skje. I tillegg, med det lave antallet datapunkter som blir brukt i analysen, kan et tilfelle hvor juicen forårsaker at brød får tildelt en større glukosefeil, lett føre til et mye mer spredt prediksjonsintervall. Dette vil justere brødskeive skalaren i feil retning.

Et annet problem som gruppen ikke hadde tenkt gjennom var om en skalar ble for lav. Grunnen til at gruppen ikke hadde tenkt på dette som et mulig problem var at

---

det ikke var forventet at algoritmen skulle endre dataene så drastisk som den gjorde. Dersom en skalar blir for lav, vil dette bety at matvaren fikk for lite insulin. Dette vil føre til en høyere glukoseverdi, altså en større feil. Ettersom matvaren ikke sto for så mye av det totale insulinet, vil den ikke få mye av skylden. Dette hadde også jevnet seg ut med mange iterasjoner, men antallet iterasjoner ville, som nevnt over, tatt lang tid å samle inn, og det er ikke aktuelt for en bruker å gå over lenger perioder med dårlig regulering av blodsukkeret.

En alternativ partisjoneringssteknikk, kunne vært å fordelt den totale feilen jevnt mellom alle partene. Dette ville hatt samme problemer som nevnt over, men siden alle får samme mengde feil, så ville dette kanskje jevnet seg ut på færre iterasjoner.

### 5.2.1.2 Insulin til glukose forholdet

Da gruppen lagde algoritmen, og valgte hvilke deler av systemet som skulle kunne endres gjennom skalarer, ble dagsprofiler et naturlig valg, da disse ble laget for dette formålet. Matvarer fikk også egne skalarer grunnet variasjon i karbohydrater [45]. Gruppen valgte i tillegg å gi korrigeringsinsulinet en egen skalar. Tanken var at det ofte blir satt korrigeringsinsulin, og at mengden som trengs også kan endre seg med tiden, og derfor ble dette lagt inn. Det gruppen overså var at det alltid vil være en dagsprofil involvert i et måltid som har en glukose-skalar, som kunne tatt hånd om dette behovet. Det var derfor dobbelt opp med håndtering av glukose sensitiviteten, og det ble også flere uventete problemer med korrigeringsskalaren.

Når korrigeringsskalaren endres, oppdateres også «insulin-til-glukose»-forholdet seg. Dette ser man i figur 14q. Det var ikke før de siste ukene at denne verdien gjennomgikk store endringer, og fram til dette så det ut som om algoritmen hadde justert verdien og var relativt sikker på denne. Problemet befant seg i metoden for analysen. Analysen ble utført på glukosefeil per enhet med korrigeringsinsulin, det vil si at en feil på X glukose som ble partisjonert til korrigeringsinsulinet på Y enheter, fikk et datapunkt med X delt på Y glukose feil per enhet korrigeringsinsulin. Dette førte til at mange av datapunktene lå nærmere X-aksen og prediksjonsintervallet var relativt bredt. Dette på grunn av partisjoneringsmetoden, som diskutert over, og lite data grunnlag, grunnet paramenterene. Dette førte til at algoritmen foreslo å gi mindre insulin, og dette gir en skalarfaktor under én. Ettersom «insulin-til-glukose»-forholdet blir oppdatert ved å dele på denne faktoren vil forholdet øke.

Da nye måltider skulle beregnes, ble mindre insulin foreslått til reguleringen, som igjen ga en større feil. Denne feilen ble da gitt delevis til korrigeringsinsulinet, og dermed får den mindre av feilen, siden den foreslo mindre insulin. Dette fører til at den fortsatt vil få datapunkter nær null i analysen som vil føre til at «insulin-til-glukose»-forholdet blir større.

«Insulin-til-glukose»-forholdet brukes også for å regne ut de teoretiske endringen som trengs i de andre skalarene. Det at dette forholdet da endres dynamisk av algoritmen, er en dårlig idé. Dette tillot algoritmen å endre og justere en av de viktigere grunnsteinene sine. Dette genererte repeterende økende verdier, som til slutt førte til den drastiske økningen vist i figur 14q.

Videre får dette dårlig påvirkning på alle data. Dette er fordi det skjer en justering av feil i hver logg når det blir partisjonert glukosefeil fra en påminnelse til de

---

tilkoblede loggene. På slutten av testperioden var «insulin-til-glukose»-forholdet på rundt 15 mmol/L per enhet. Testpersonen vil, i hvert tilfelle, alltid justere estimatet til nærmeste halve enhet, men testpersonen overskriver estimatet dersom vedkommende er uenig. Dette skjedde oftere mot slutten, da den foreslåtte korrigeringsdosen var alt for liten. Om testpersonen foreslo én enhet ekstra til et måltid og oppnådde ønsket blodsukker, vil feilen i påminnelsen være null, men i loggen som hører til, vil den bli etterjustert for denne ene enheten. Justeringen blir beregnet ved hjelp av «insulin-til-glukose»-forholdet som på slutten ville resultert i at loggen hadde en feil på da 15 glukose (mmol/L). Dersom testpersonen hadde endret med to enheter, hadde det vært 30 ( $2 * 15$ ). Dette førte til en ekstrem forstørring i variasjonen i dataene, som medførte at prediksjonsintervallene ble mye bredere, som kan ha ført til den generelle synkingen i skalarverdier i alle tilfeller.

Om algoritmen ikke hadde endret på «insulin-til-glukose»-forholdet, men latt dette ansvaret falle på glukose-skalarene i dagsprofilene, kan det hende at resultatene hadde vært bedre. Uansett, har testpersonen gjennom hele prosessen justert dosene til det vedkommende var komfortabel med, eller sikker på.

### 5.2.1.3 Overlappende måltider

Overlappende måltider er når et nytt måltid blir loggført før påminnelsen til det første har nådd sluttiden sin og er ferdig. I disse tilfellene vil disse måltidene kobles sammen opp mot samme påminnelse, og påminnelsen forlenges. Gruppen gjorde en antagelse på at insulinmengden til det første måltidet ble beregnet korrekt. Man bruker da ønsket glukoseverdi i første måltid som nåværende glukoseverdi i andre måltid. Denne metoden fungerer bra for måltider som overlapper nært hverandre, som for eksempel middag og dessert. De kunne nesten blitt lagt inn som et og samme måltid. Det er derimot i tilfeller hvor måltidene akkurat overlapper at dette blir problematisk. Et scenario er hvor første måltid nesten er ferdig og glukoseverdien er på 15,0 mmol/L, og et nytt måltid blir lagt inn. Dersom det andre måltidet er perfekt dosert, vil glukoseverdien for påminnelsen likevel være 15,0 mmol/L. Feilen vil bli beregnet og fordelt mellom loggene basert på bidrag. Dette blir utført som forventet, og er bra fra et analytisk perspektiv. Dette resulterer derimot også med at testpersonen har hatt høye glukose-verdier gjennom hele den andre perioden, som fører til et dårligere regulert blodsukker. Dette problemet ble ikke adressert i den nåværende løsningen. Det kunne derimot vært lagt inn funksjonalitet, der man kunne lagt inn, ved et overlappende måltid, at man satte korrigeringsinsulin som burde blitt satt i forrige logg. På denne måten kunne man rettet opp i feil man, som menneske, kjenner igjen. For eksempel, enhver diabetiker vet at dersom man er 10 mmol/L over ønsket glukoseverdi, og det er 15 minutter igjen av aktivt insulin, vil man ikke til å oppnå ønsket verdi. Dermed kan man allerede på dette tidspunktet sette mer insulin.

Problemet for testpersonen var dersom man økte mengden insulin satt til det andre måltid for å korrigere den for høye glukoseverdien fra første måltid. Algoritmen ville tolket dette som om testpersonen satte mer insulin til matvarene i andre måltid. Dette vil bli feilaktig data for algoritmen å bruke, som kan resultere i at de involverte matvarene blir justert feil senere. Testpersonen valgte i enkelte tilfeller å vente til påminnelsen gikk ut før de la inn det nye måltidet, for å unngå dette problemet. I enkelte tilfeller satte testpersonen insulin manuelt, og lot være å loggføre det. Dette for å unngå å sette feilaktige doser som kunne ført til at algoritmen justerte seg

---

feil, men også for å sikre at blodsukkeret ikke var høyt for lenge, for å forhindre helseproblemer, se delseksjon 2.1.

#### **5.2.1.4 Skalaren for korrigeringsinsulin uten matvarer**

Det ble laget en egen skalar som ble brukt dersom bruker kun satte korrigeringsinsulin. Denne var ment for å forklare endringen i sensitivitet når testpersonen ikke hadde spist på en stund. Denne ble aldri brukt da testpersonen alltid la inn matvarer. Gruppen innså at denne skalaren var for simpel til å forklare dette komplekse tilfellet, da denne skalaren ville blitt brukt om man loggførte en eksklusiv korrigeringsdose rett etter et måltid. Skalaren burde variere basert på tid siden siste måltid, og potensielt andre faktorer. Derfor valgte gruppen, da de implementerte algoritmen, å ikke la denne skalaren oppdatere seg som de andre skalarene. Gruppen fjernet heller ikke variabelen, da dette ble sett på som unødvendig bruk av tiden grunnet at testpersonen som nevnt tidligere, var en av utviklerne, og satt bare denne verdien til én. På denne måten hadde ikke skalaren en innvirkning dersom den skulle blitt tatt i bruk.

### **5.2.2 Presisjon**

Presisjonen til algoritmen måles gjennom HbA1c-verdien, slik som nevnt i 3.2.7. Som observert i underfigur 16a, var HbA1c-verdien 7,5 prosent, eller 58 mmol/mol, ved prosjektets slutt. Til sammenligning var testpersonens HbA1c-verdi 7,3 prosent (56 mmol/mol) før MVP-en og algoritmens introduksjon. Som nevnt i delseksjon 2.1 vil denne verdien for en person uten diabetes ligge under 42 mmol/mol eller 6 prosent, og for en diabetiker er behandlingsmålet 53 mmol/mol, eller 7 prosent.

Presisjonen til algoritmen nådde ikke det som var målet. Basert på det som ble diskutert i delseksjon 5.2.1.2, er det relativt sikkert at algoritmen vil fortsette å øke HbA1c-verdien. Fra figur 16a, viser ITS analysen at denne vil fortsette å stige. Dette støtter diskusjonen så langt. Det bekrefter svakheten i algoritmen, altså «insulin-til-glukose»-forholdet, og hvordan dette dynamisk endres. Dette burde vært implementert på en annen måte. Selv om testpersonen endrer verdiene, vil algoritmen i dens tilstand ved slutten av testperioden, gjøre en dårligere jobb enn tidligere i testperioden.

### **5.2.3 Parametere til algoritmen**

#### **5.2.3.1 Minste antall datapunkter**

Som nevnt i delseksjon 3.3.1.3, ble parameterene for algoritmen valgt basert på grundig undersøkelse av hva som kunne være fornuftige tall. For eksempel ble parameteren «minste antall tilfeller» satt til ti. Dette var på bakgrunn av en forskningssartikkel, som sa at den generelle anbefalingen er å ta i bruk tolv datapunkter før og etter en avbrytelse i en ITS analyse. Grunnet for lite tid, hadde ikke gruppen mulighet til å undersøke ulike tallverdier for denne parameteren. Å ha ulike parameterverdier for algoritmen gjennom testperioden ville resultert i for lite data for de forskjellige parameterverdiene. Parameterne påvirket definitivt resultatene, men gruppen konkluderte med at dersom algoritmen skulle være praktisk brukbar for en

---

testperson, måtte iterasjonene skje med minst mulig data. Gruppen anså at dersom algoritmen gjennomførte nok iterasjoner, ville den kommet frem til omtrentlig de samme resultatene, som den ville dersom iterasjonene hadde flere datapunkter. Dette grunnet at algoritmen med hver iterasjon vil forbedre seg, og glukosefeilen vil nærme seg null. Problemet med dette var skjørheten til dataene og prediksjonsintervallet, grunnet det begrensede antallet datapunkter. Hadde det vært et større datagrunnlag per iterasjon kunne analysene vært mer robuste og hatt smalere prediksjonsintervall. Med ti datapunkter, og når man tar i betraktning det som hendte med «insulin-til-glukose»-forholdet beskrevet i delseksjon 5.2.1.2, ble det for lite data for å få snevre prediksjonsintervall.

### **5.2.3.2 Nedre grense for prediksjonsintervall**

Den nedre grensen for prediksjonsintervallet ble bevisst satt litt konservativt for å beskytte testpersonen mot for høye doseringer. Dette for å unngå svært lave verdier, og potensielt følinger. Dette har skjeldent vært et problem, og i de tilfellene det var problematisk, kunne testpersonen håndtere dette med god tid. Basert på erfaringen fra testperioden, kunne det vært mulig å sette denne verdien enda lavere.

Dersom verdien hadde vært lavere, kunne det hende at resultatene hadde vært en del anderledes. Med bakgrunn i diskusjonen i delseksjon 5.2.1.2, var mest sannsynlig denne parameteren en sentral del av økningen i «insulin-til-glukose»-forholdet. Dette førte til at glukoseverdiene gikk opp og skalarene gikk ned, som stemmer overens.

### **5.2.3.3 Absolutt største justeringsfaktor**

Parameteren «absolutt største justeringsfaktor» har direkte vært involvert i nesten alle iterasjonene av algoritmen. Det kan hende at det er denne parameteren som sikret at «insulin-til-glukose»-forholdet ikke endret seg for fort før slutten av testperioden. Dette er mest sannsynlig også tilfellet med resten av skalarene, som grunnet den store variasjonen i slutten, ville ønsket å endre seg mye mer enn det de faktisk gjorde, se delseksjon 5.2.1.2. Det var nettopp dette som var hensikten med parameteren, det å begrense endringer som er forårsaket av prediksjonsintervall som har delevis korrupt data i dem, slik at endringer som anbefales på grunn av enkelte ekstremalverdier, ikke skjer i sin helhet. Derimot om det er en trend, vil det bli anbefalt å endre skalaren i samme retning flere ganger, og den totale endringen vil skje over flere iterasjoner.

### **5.2.4 Insulinets virketid**

Som nevnt i delseksjon 2.1.1, har insulin ofte en maksimal effekt i én til tre timer etter at det først har blitt satt. Insulin kan fortsatt være aktivt i kroppen i tre til fem timer etter at det ble satt. Gruppen valgte å bruke blodsukker verdien testpersonen hadde tre timer etter insulin ble satt. Blodsukkeret blir da ansett som fastende. Dette betyr at verken mat eller insulin påvirker blodsukkeret lenger, da dette gir det mest gyldige målet opp mot ønsket oppnådd verdi. Denne verdien har fungert bra. Testpersonen har som regel mer enn tre timer mellom måltidene sine, som medfører at det ikke er så mange overlappende måltider og derav mer korrekt data. Det er også verdt å nevne at

---

denne verdien ble satt på grunnlag av den spesifikke insulintypen testpersonen bruker. Det vil si at denne verdien burde variere fra tilfelle til tilfelle, basert på insulintypen dersom en større testgruppe deltar i en ny testperiode.

## **5.3 Oppnådde mål**

### **5.3.1 Korrelasjonsmatriser**

Denne delseksjonen tar for seg korrelasjonsmatrisene i delseksjon 4.1.1.5. Man kan lære mye av hvordan algoritmen og MVP-en har påvirket reguleringen av blodsukkeret til testpersonen basert på disse. Selv om resultatet av reguleringen, i form av HbA1c-verdien og variasjon oppnådd, ikke har endret seg stort, kan man observere om sammenhengen mellom de forskjellige delene av reguleringen, henger sammen på samme måte, eller om noe har endret seg.

#### **5.3.1.1 Gjennomsnittlig glukose**

Forholdet til HbA1c-estimatet og veldig høye glukoseverdier har ikke endret seg i noen stor grad. Derimot, har forholdet til glukose variasjonen, høye glukoseverdier, ønskede glukoseverdier og veldig lave glukoseverdier har endret seg opp til 20%-poeng, dette er en stor endring, men korrelasjonen har ikke snudd, eller blitt neglisjerbar. Dette kan dermed være naturlig variasjon. Forholdet til lave glukoseverdier derimot har endret seg fra -0,60 til -0,90. Dette er 30%-poeng endring og er nå blitt en veldig sterk negativ korrelasjon. Dette vil si at ved bruk av MVP-en, vil en økning i gjennomsnittlig glukose nesten definitivt gi en reduksjon i lave glukoseverdier. Dette kunne man også observere tidligere, før MVP-en, men nå er det svært sikkert har dette vil skje.

#### **5.3.1.2 HbA1c-estimat - GMI**

Siden HbA1c-estimatet og gjennomsnittlig glukose har en positiv korrelasjon på 0,99 både før og under MVP-en, ser man at resultatene her er identiske til de diskutert om gjennomsnittlig glukose.

#### **5.3.1.3 Glukose variasjon**

Forholdet til veldig høye glukoseverdier er tilnærmet uendret. Høye glukoseverdier har gått litt ned, men har fortsatt samme betydning av korrelasjonen. Ønskede glukoseverdier er derimot gått ned 21%-poeng, og har ikke lenger en sterk nok korrelasjon til at det kan anses som sikkert. Dette er positivt, da korrelasjonen betydde at om tiden med ønskede glukoseverdi økte så økte også variasjonen. Nå er det ikke like sikkert at dette er tilfelle. Optimalt ønsker man all tiden innenfor ønskede glukoseverdier, og en veldig lav variasjon. Optimalt, og teoretisk, ønsker man en negativ korrelasjon mellom disse, noe den nå er næremere. Lave glukoseverdier og veldig lave glukoseverdier sitt forhold til variasjonen har økt med henholdsvis 27%-poeng

---

og 35%-poeng, og er nå 0,63 og 0,53. Det er nå en betraktelig større sammenheng mellom økningen i variasjon og økningen i lave og veldig lave glukoseverdier.

#### **5.3.1.4 Veldig høye glukoseverdier**

Forholdet til høye glukoseverdier har økt fra -0,11 til 0,20, dette er 31%-poeng og et skifte i fortegn. Det er fortsatt ikke en sterk sammenheng, men den store endringen og spesielt fortegnsskifte er interessant å notere seg. Forholdet til de ønskede glukoseverdiene har hatt en økning på 31%-poeng, fra -0,31 til -0,62, som er en dobling. Dette er en god utvikling. Det er fordi det nå er sikrere at dersom veldig høye glukoseverdier går ned, vil de ønskede gå opp. Dette kan tyde på at tiden i sonen med veldig høye verdier, migrerer direkte til de ønskede verdiene i større grad. Forholdet til de lave glukoseverdiene har ikke endret seg stort. Forholdet til de veldig lave glukoseverdiene derimot har gått fra -0,58 til -0,27, en reduksjon på 31%-poeng. Der er ikke lenger en like sikker sammenheng mellom økningen i veldig høye verdier, og reduksjonen i veldig lave verdier. Dette kan komme av flere årsaker, men det er vanskelig å si hva som har forårsaket dette.

#### **5.3.1.5 Høye glukoseverdier**

Forholdet til ønskede glukoseverdier og veldig lave glukoseverdier har ikke-interessante endringer. Forholdet til lave glukoseverdier har derimot en veldig stor endring, fra 0,02 til -0,75, 77%-poeng forskjell. Fortegnsskifte er ikke så interessant da den første verdien lå så nære null, men den store endringen fra ingen korrelasjon til betydelig korrelasjon er meget interessant. Under MVP-en var det altså en sammenheng mellom økningen av høye verdier, og reduksjonen av de lave verdiene. Om man ser for seg at man har et intervall mellom to glukoseverdier hvor de fleste av dagenes glukoseverdier havner innen for dette. Før MVP-en, kan dette intervallet ha vært stort nok til å inneholde både høye og lave glukoseverdier, derav ingen korrelasjon. Dersom intervallet ble smalere, slik at enten høye verdier eller lave verdier var innenfor (altså mindre variasjon), ville dette forklart den plutselige økningen i korrelasjon.

#### **5.3.1.6 Ønskede glukoseverdier**

Forholdet til veldig lave glukoseverdier er nesten uendret. Derimot forholdet til lave glukoseverdier har i likhet med forholdet mellom høye og lave glukoseverdier, en veldig stor endring. Denne endringen er fra -0,08 til 0,73, 81%-poeng forskjell. Her er heller ikke fortegnsskifte interessant, grunnet at første verdi lå nær null. Her kan man se at ved bruk av MVP-en, var det tydelig at om de ønskede glukoseverdiene økte, økte også de lave glukoseverdiene. Dette er ikke nødvendigvis en bra endring. Det kan tolkes som at dersom man først er innenfor de ønskede verdiene, er sjansen for å oppleve lave glukoseverdier stor. Det kan også tolkes som at man oftere ligger i det lavere sikte av de ønskede verdiene, og dermed oftere opplever lave glukoseverdier.

---

### 5.3.1.7 Lave glukoseverdier

Forholdet til de veldig lave glukoseverdiene har gått fra 0,61 til 0,43. Denne sammenhengen er altså svekket. Dette er en god endring. Grunnet til dette er at dersom mengden tid med lave verdier øker, vil ikke nødvendigvis tiden med veldig lave verdier også øke. Dette kan tolkes som at man sjeldnere vil oppleve at lave glukoseverdier utvikler seg til veldig lave verdier, med bruk av MVP-en.

### 5.3.1.8 Veldig lave glukoseverdier

Forholdet til veldig lave glukoseverdier har blitt diskutert gjennom de tidligere delseksjonene over.

## 5.3.2 Algoritmen

Fra resultatene i delseksjon 4.1.1.4, presentert i underfigur 16a, kan man observere at HbA1c-verdien til testpersonen har en stigende trend, i stedet for å synke slik som målet var. Det er også et ekstremalpunkt i slutten av dataen og en relativt lite datasett, som kan ha blitt påvirket av ekstremalpunktet. Ved introduksjonen av MVP-en, kan man observere med ITS analyse, at HbA1c-verdien sank før den begynte å stige. Det var forventet at denne verdien ville bli dårligere helt i starten ved testperioden, da det først bare var MVP-en og algoritmen trenger noen iterasjoner for å forbedre seg når den først er i gang. Man kan imidlertid observere at den faktisk gikk litt ned, eller holdt seg nede, før den begynte å stige, og den har ikke forbedret seg. Det kan være at de teoretiske verdiene MVP-en alene, før algoritmen, foreslo var relativt nærme de faktiske verdiene. Dersom man fjerner det siste punktet som ble mest påvirket av effekten diskutert i 5.2.1.2, ville linjen vært relativt horisontal, da man ser at dataene ligger godt samlet, bedre enn før testperioden. I tillegg er det en generell bedre samling av datapunktene etter introduksjonen av MVP-en og algoritmen. En mulig grunn til at HbA1c-verdien steg, er at algoritmen ikke har fått nok tid til å justere seg. Dette er lite trolig da nesten alle skalarverdier har sunket, som vil føre HbA1c-verdien oppover. Det kan mest sannsynlig være at algoritmen ikke fungerer slik som den har blitt implementert nå, det som ble diskutert i delseksjon 5.2.1.2. Algoritmen oppfører seg eller som forventet.

Som vist i delseksjon 4.1.1.4 i underfigur 16c, økte blodsukkerets variasjon etter introduksjonen av MVP-en. Dette ser gruppen på som forventet oppførsel, da algoritmen trenger litt tid, og iterasjoner, for å tilpasse og justere ulike skalarer. Ved å justere disse skalarene basert på innsamlet data, vil algoritmen bli bedre til å estimere insulinmengder. Man kan observere i underfiguren at ITS analysen viser en nedgang mot det ønskede intervallet for variasjon i blodsukkeret. Ved slutten av testperioden hadde testpersonen en uheldig dag, 29.04.2022, der blodsukkeret steg til over 27 mmol/L mens han trente. Natten deretter lå også blodsukkeret under tre mmol/L hele natten. Gruppen tror dette bidro til svært mye variasjon, som kan forklare hvorfor det nest siste punktet i underfiguren var på over 44%. Til tross for denne ekstra variasjonen, sier fortsatt analysen at variasjonen har en synkende trend. Det er interessant å påpeke at det er en korrelasjon på  $-0,71$  mellom variasjonen og tiden i sonen for høye blodsukkerverdier, se tabell 3. Som man tydelig kan se i de respektive grafene i figurene 16c og 15d. Når variasjonen øker, så synker tiden i det høyest



---

blodsukkeret og motsatt. Variasjonen har også  $-0,55$  og  $-0,52$  i korrelasjon respektivt med gjennomsnitts glukose og HbA1c-estimatet. Det vil si at desto høyere verdier blodsukkeret ligger på, desto mindre variasjon er det. Dette kan komme av en såkalt «berg-og-dal-bane» effekt. Dette går ut på at dersom blodsukkeret blir for lavt, vil både kroppen og testpersonen, ved å konsumere mat, forsøke å øke blodsukkeret. Ofte kan det resultere i en overkorrigering som gjør blodsukkeret høyt. Deretter vil testpersonen gi insulin for å få ned blodsukkeret igjen. Dette kan skje gjentatte ganger. Blodsukkeret går da opp og ned som en berg-og-dal-bane. Denne effekten skaper mye variasjon. I tilfellet hvor blodsukkeret er på en litt høyere verdi, vil det ikke være like lett å få denne effekten og variasjonen vil være mindre. Dette forholdet har også blitt svekket litt ved bruk av MVP-en og algoritmen som diskutert i delseksjon 5.3.1.

Tiden innenfor de ulike sonene for blodsukkeret har endret seg i løpet av testperioden. Som vist i underfigur 15d kan man se at sonen for veldig høyt blodsukker har en økende trend under testperioden. Dette indikerer at algoritmen har foreslått for små insulindoser som gir et for høyt fastende blodsukker etter et måltid. Man kan også observere at testpersonen har vært omtrent like mye i sonen for høyt blodsukker før og under testperioden. Gruppen tror den økende trenden har en sammenheng med sikkerhetsmekanismen beskrevet i delseksjon 3.2.5, og at testpersonen har vært mindre i sonen for lavt blodsukker. Her har korrelasjonen mellom høye og lave glukoseverdier en interessant utvikling. Før var den  $0,02$ , se tabell 2, og  $-0,75$  etter, se tabell 3. Det er tydelig at ved bruk av MVP-en og algoritmen, er det en klar sammenheng mellom økningen i høye verdier og reduksjonen i lave verdier. Man kan tenke seg at tiden med lave verdier har migrert opp i ønskede verdier, og ut igjen i de høye verdiene, noe som ikke var tilfellet før testperioden. Basert på figurene og resultatene fra resultatanalysen, er gruppens hypotese at algoritmen har forskjøvet store deler av glukoseverdiene opp, og at den ikke har klart å regulere blodsukkeret på en måte der man unngår svært høye blodsukkerverdier, dette med bakgrunn i delseksjon 5.2.3. Dette støttes også opp av korrelasjonen mellom høyt blodsukker og ønsket blodsukker sonene, som er på  $-0,85$  før og etter testperioden, se tabell 2 og 3. Som man kan observere, øker antall tilfeller av høyt blodsukker, og ønsket blodsukkerverdier har gått ned. Det er svært tydelig at data forflytter seg i stor grad fra ønsket sone til høy sone, gitt den høye korrelasjonen. Det virker altså som blodsukkeret har blitt forskjøvet opp.

Man kan observere at tiden innenfor ønskede blodsukkerverdier jevnt har sunket, men at det ikke er en drastisk forskjell fra tidligere, før det aller siste datapunktet. Gruppen mener at algoritmen har forskjøvet deler av tiden innenfor de ønskede nivåene opp til sonen for høyt blodsukker, og at enkelte av disse verdiene igjen har blitt forskjøvet til sonen for veldig høyt blodsukker. Gruppen tror dette kommer av den nedre grensen, se delseksjon 3.2.5. Det er viktig å påpeke at sonen for høyt blodsukker er svært snever, og går kun fra  $10,1$  til  $13,9$  mmol/L. Dette til forskjell fra for eksempel veldig høyt blodsukker, som er alle verdier over  $13,9$  mmol/L, eller den ønskede sonen som varierer fra  $3,9$  til  $10,0$  mmol/L. Det at blodsukkeret har forskjøvet seg opp støttes også av den synkende trenden i lavt og veldig lavt blodsukker.

En annen sone som også er ekstrem snever er lave blodsukkerverdier. Denne sonen er fra  $3,0$  til  $3,8$  mmol/L. Da MVP-en ble introdusert kan man observere en økning i antall tilfeller i denne sonen, slik at testpersonen brukte over 4% i denne sonen, og dette er ikke ønsket. Etter litt tid begynte denne å ha en kraftig nedgang, slik at testpersonen igjen var innenfor den ønskede tiden for denne sonen. På slutten av

---

testperioden var tiden innenfor denne sonen med en sterkt synkende trend. Dette kan være et resultat av den nedre grensen i algoritmen, som har forskjøvet deler av tilfellene opp i sonen for ønskede blodsukkerverdier. Om man ser på korrelasjon mellom ønsket blodsukker, og lavt og veldig lavt blodsukker, henholdsvis -0,08 og 0,06 før testperioden og 0,73 og -0,05 under, se tabell 2 og 3. Det er ikke en sammenheng mellom ønskede og sonen for veldig lave verdier. Det er derimot skjedd en større endring mellom ønskede og lave verdier. Det var ingen sammenheng før, men under testperioden er det en relativt sterk korrelasjon som sier at, dersom ønskede verdier øker, vil lave verdier også øke. Som diskutert i delseksjon 5.3.1, så kan det være at doseringen til algoritmen oppnår verdier i det lavere sikte av de ønskede verdiene. Det vil da oftere kunne skje at glukoseverdiene til testpersonen naturlig migrerer ned i lave verdier grunnet variasjon.

Dette kan være fordi forskyvingen av verdier fra de to nederste sonene vil føre til at tilsvarende mengde verdier migrerer av ønsket sone og opp i de to øverste sonene. For sonen for høyt blodsukker ser man, som nevnt tidligere, en negativ korrelasjon med lave verdier. For veldig høye blodsukkerverdier ser man også en negativ korrelasjon til både lave og veldig lave verdier. Dette opptrer både før og etter testperioden, se tabell 2 og 3. For sonen for veldig høye verdier, er korrelasjonen henholdsvis -0,60 og -0,27 i testperioden. Sonen for ønskede glukoseverdier forblir relativt uendret, og det var kun en svak synkende trend. Tiden brukt i sonene for lave og veldig lave glukoseverdier går ned, og tiden brukt i sonen for høye og veldig høye glukoseverdier går opp. Med disse observasjonene kan man antyde at det migrerer omtrent like mye tid fra de lave og veldig lave sonene, inn i den ønskede sonene, som det går ut av den ønskede sonen og over til de høye og veldig høye sonene. Dette støttes videre av den negative ikke-neglisjerbar korrelasjon mellom de to nederste sonene (lave og veldig lave) og de to øverste (høye og veldig høye).

Dersom man undersøker sonen for veldig lave blodsukkerverdier, kan man observere at det var en økning ved MVP-en sin introduksjon, men at denne også har avtatt. Som man kan se i figuren for denne sonen, 15a, var det ingen tilfeller av veldig lave blodsukkerverdier i perioden fra omkring 17.11.2021 til 07.01.2022. Dette betyr ikke nødvendigvis at testpersonen var godt regulert, og dette kommer godt frem, spesielt i figuren for høye blodsukkerverdier, 15d, men også i figuren for ønskede blodsukkerverdier, 15c. Som man kan se hadde testpersonen flere høye verdier i sonen for høyt blodsukker i denne perioden, samt at sonen for ønsket blodsukker hadde flere lave verdier. Dette gjenspeiles igjen i korrelasjons verdien mellom disse sonene, 1. Dersom man sammenligner med tidsperioden etter MVP-en sin introduksjon kan man se likhetstrekk mellom denne perioden og de siste innsamlede datapunktene.

Algoritmen har justert de ulike skalarene for både matvarer, dagsprofiler og korrigeringsinsulin. Matvarene «Yoghurt», «Appelsinjuice», og «Bringebærsyltetøy» har ikke gjennomgått spesielt mange iterasjoner, og det er vanskelig å trekke konklusjoner på disse. For matvaren «Brød» kan man se antydninger til at skalaren for denne begynner å finjustere seg rundt 0,6 før «insulin-til-glukose»-forholdet endret seg drastisk, se delseksjon 5.2.1.2. I tillegg til at man kan observere at glukosefeilen har samlet seg litt mer rundt 0 mot slutten av testperioden, enn før og rett etter algoritmens introduksjon.

Den generelle trenden for alle dagsprofilene er at skalarene har gått ned fra startverdiene sine. Dette skyldes mest sannsynlig, som diskutert tidligere, den nedre grensen for prediksjonsintervallet, se delseksjon 5.2.3. Dette var også en trend før effekten av

---

«insulin-til-glukose»-forholdet begynte å påvirke dataene, se delseksjon 5.2.1.2. I tiden etter dette ser man mye av det samme, men i enkelte tilfeller kan man se at skalarene faktisk går opp. Dette er kun glukose-skalarene. Gruppen tror dette kommer av at det blir estimert mye mindre korrigeringsinsulin, som fører til at feilen er liten nok til at skalaren kan øke. Det er også gjort større endringer enn tidligere som kan tyde på at den store variasjonen, grunnet «insulin-til-glukose»-forholdets økning, kan ha gjort skalarendringene på dagsprofiler større.

Korrigeringsskalaren har alt blitt nevnt og diskutert, og den har sin egne delseksjon 5.2.1.2.

### **5.3.3 Resultat- og effektmål**

Som nevnt i delseksjon 5.3.2, steg HbA1c-verdien til testpersonen. Resultatmålet om å senke denne verdien med to prosentpoeng har altså ikke blitt oppnådd. Grunnen til at denne verdien ikke reduserte seg, kan være at algoritmen ikke hadde nok tid til å justere de ulike skalarene for et best mulig resultat. En annen grunn til dette resultatet kan være at den nedre sikkerhetsgrensen medførte at algoritmen justerte insulinmengdene slik at blodsukkeret steg vekk fra de lave verdiene, slik som diskutert i delseksjon 5.3.2.

Som presentert i delseksjon 4.2.1.1, ble mengden tid brukt på å regulere blodsukkeret i uken redusert fra omtrent 181 til 108,8 minutter i uken (gjennomsnittlig). Dette tilsvarer en reduksjon på omtrent 40%. Resultatmålet om å redusere tiden brukt på å regulere blodsukkeret med 50% ble altså ikke oppnådd, men dette var et veldig ambisiøst mål. Målet henger tett sammen med effektmålet om å gjøre det lettere og mindre tidkrevende å regulere blodsukkeret. MVP-en gjør det mulig for diabetikere å lagre matvarer, og informasjon om disse, de spiser i en database. Man kan søke etter matvarer, og enkelt legge de til i en liste over matvarer man ønsker å konsumere ved et måltid. Ved å lagre denne informasjonen, og ha den lett tilgjengelig gjennom en mobil enhet, er mest sannsynlig det som har bidratt mest til at det har vært lettere og mindre tidkrevende for testpersonen å regulere blodsukkeret. Dette er fordi ved å ha denne informasjonen lagret lett tilgjengelig, er det ikke lenger behov for å konstant undersøke hvor mange karbohydrater en matvare har, mer enn den ene gangen man legger matvaren inn i MVP-en. For testpersonen var det lettere å vite hvor mange enheter de normalt satt til en matvare de spiste hver dag, som for eksempel brød, enn å huske nøyaktig hvor mange karbohydrater det var. Denne metoden åpner ofte opp for feildosering, da dette baserer seg på minne til testpersonen, og hva de er vant til å sette, og ikke nødvendigvis hva de burde sette. I tillegg var testpersonen nødt til å undersøke enhver matvare der vedkommende ikke husket hvor mange enheter de normalt satte. Slike tilfeller oppstår som regel ikke så ofte i hverdagen, men det kan fortsatt ta tid, samt at det er tungvint. Andre aspekter som kan ha bidratt til at tiden brukt på å regulere blodsukkeret gikk ned, er at MVP-en kalkulerer en insulin dose for testpersonen. Testpersonen må dermed ikke selv undersøke antall karbohydrater, og regne ut dette selv, hver gang de skal konsumere en matvare de ikke husker denne informasjonen for. De må heller ikke legge sammen karbohydrater for ulike matvarer. Til tross for at testpersonen alltid verifiserer at doseringen er trygg så går det kjappere med kalkulatoren i MVP-en, enn om de skulle regnet ut dette selv.

Det er svært mange aspekter i hverdagen som påvirker diabetes, som trening, ulike typer karbohydrater, dagsform og så videre. Ettersom det er svært få av disse

---

faktorene som er nøyaktig like hver dag, kan det være svært vanskelig å regulere blodsukkeret. Det skaper også en utfordring ved algoritmen. Det å undersøke om det faktisk er algoritmen som har bidratt til et bedre regulert blodsukker, eller om det er eksterne faktorer som har bidratt til dette, er ikke trivielt. Som observert i resultatene fra resultatanalysen, delseksjon 4.1.1, viser ikke analysen at testpersonen har blitt bedre regulert. Likevel kan man observere at testpersonen ikke nødvendigvis har blitt dårligere regulert heller. Ved å se bort ifra tiden etter «insulin-til-glukose»-forholdet gikk drastisk opp, se delseksjon 5.2.1.2, så kan det nesten se ut som blodsukkeret var like godt regulert som før. Som man kan se har HbA1c-verdien økt, og dette er verdien helsepersonell bruker for å vurdere hvor godt regulert en diabetiker er. Likevel kan man observere at HbA1c-verdiene under testperioden har vært lavere, med unntak av siste verdi, enn før testperioden, de er også mer samlet og det er mindre spenn i punktene. Dette er da kun om man ser bort ifra den siste verdien. Dette indikerer at selv om HbA1c-verdien har økt, og testpersonen har blitt litt dårligere regulert enn det vedkommende var rett før MVP-en sin introduksjon, har ikke testpersonen blitt like dårlig regulert som det vedkommende har vært tidligere. Dette representeres da av de første punktene i figuren 16a. HbA1c-verdien er en verdi som blir målt ofte fire ganger i året, for å undersøke godheten av reguleringen. Mot slutten av testperioden begynte testpersonen å trene mer, og som nevnt i delseksjon 2.1.1, vil trening påvirke insulinsensitiviteten. Dette kan ha bidratt til at HbA1c-verdien ikke hadde en nedgang, da algoritmen ikke tar hensyn til slike hendelser.

Designet ble utarbeidet med brukervennlige prinsipper, som for eksempel visuell konfirmasjon, og synlig og konsistent funksjonalitet. Fra brukertestene kom det frem at blant annet navnet «Dagsprofil» kan være noe misvisende, men ved nærmere undersøkelse forsto brukerne hva funksjonaliteten omhandlet. Etersom brukertestene ble utført relativt sent i prosessen, valgte gruppen å beholde navnet, slik at kildekoden skulle ha en sammenheng med designet. I tillegg har navnet «Dagsprofil» blitt brukt i alt av dokumentasjon. Dette er imidlertid et ord gruppen vil bytte for at funksjonaliteten skal bli bedre presentert. Det har vært viktig for gruppen å utarbeide designet på en best mulig måte, men på så kort tid som mulig. Grunnen til at gruppen ville utarbeide et intuitivt design, var på grunn av resultat- og effektmålene gruppen satte seg ved prosjektets start, da spesielt målet om å redusere tiden diabetikere bruker på å regulere blodsukkeret i uken. Da et intuitivt og godt design hegner sammen med hvor effektiv og lettbrukt produktet er. Fra brukertestene kom de også frem at de begge mente at designet var lett å bruke, intuitivt og estetisk pent å se på. Testpersonen har heller ikke opplevd problemer med designet, men ettersom testpersonen har vært med på å utvikle dette, var det viktig for gruppen å gjennomføre noen brukertester. Dette er fordi testpersonen kunne ha et bias, og vedkommende var klar over hvordan all funksjonaliteten opererte.

Etter å ha fullført testperioden med MVP-en, uttrykket testpersonen at de ønsket å forsette å bruk MVP-en. Dette er fordi testpersonen har opplevd å bruke kortere tid, samt mindre innsats på å regulere blodsukkeret i løpet av testperioden. Det skal merkes at de siste ukene hvor «insulin-til-glukose»-forholdet endret seg veldig så ble det mindre lettvent. Dersom de nødvendige endringene blir gjennomført, som å ha «insulin-til-glukose»-forholdet satt til en fast verdi, vil testpersonen fortsette å bruke MVP-en.

---

### 5.3.4 Ikke-funksjonelle egenskaper og krav

Gruppen hadde originalt planlagt å ta i bruk TestFlight for å gjøre MVP-en tilgjengelig på den mobile enheten til testpersonen. Grunnen til at dette endret seg er fordi gruppen fant en lettere løsning, som kunne gjennomføres med en gang gjennom Visual Studio for Mac. For å kunne lansere en beta versjon med TestFlight var MVP-en nødt til å gjennomgå en vurdering og undersøkelse av Apple. Gruppen visste ikke hvor lang tid dette ville ta, og begynte å undersøke alternativer, da gruppen ville unngå en forsinkelse av datainnsamling. Gruppen benyttet seg av et AdHoc-sertifikat som gjorde at MVP-en kunne lastes inn på testpersonens mobile enhet rett fra Visual Studio for Mac. Ulempen med AdHoc er at den mobile enheten må kobles til en maskin som kan overføre MVP-en og det er bare gyldig på en spesifikk mobil enhet. Dette var heldigvis ikke et hinder for gruppen. Dette siden det bare var en testperson og det var en av gruppens medlemmer.

Loggingen av systemet ble ikke gjennomført i den grad at et fullt loggingsystem ble implementert. Det var altså ikke flere nivåer med forskjellige grad av varsel. Det var derimot Debug informasjon ved de kritiske delene av koden under utvikling, noe har blitt fjernet etter at delene av systemet ble ferdig. Dette ble brukt en del under utviklingen av algoritmen. Gruppen så det ikke som nyttig å bruke et fullt logging system. Dette da de utvikler en MVP, dette skal altså være et «proof of concept» som vil si at systemet ikke nødvendigvis skal brukes videre eller utvikles videre. Systemet ville heller ikke i løpet av utviklingsprosess nå en størrelse som gjorde det vanskelig å finne bugs uten loggingen heller. Det ble derfor ikke ansett som nyttig bruk av tiden å legge inn dette.

## 5.4 Funksjonalitet i frontend

Grunnen til at målene med lavere prioritetsgrad, som events, tidsprofiler og oppkobling av MVP-en mot en sensor, ble forkastet var fordi gruppen ikke hadde nok tid til å implementere funksjonaliteten. Visjonsdokumentet ble skrevet med videreutvikling i tankene. Fra den originale oppgaveteksten ble det spesifisert minimumsfunksjonalitet. Denne besto av kalkulatoren, dagsprofiler, og matvarer som skulle legges i lister. Det ble også spesifisert at «matretter» skulle være en del av implementasjonen. Dette går da ut på at dersom man spiser en type frokost hver dag, skulle man kunne lage en «frokost entry», med for eksempel to brødskriver og et appelsinjuice glass. Grunnen til at denne funksjonaliteten ble sløffet var at etter testpersonen hadde fått brukt MVP-en med kun funksjonaliteten for matvarer implementert, opplevde de at det gikk svært raskt å finne de nødvendige matvarene. Testpersonen så dermed ikke et stort behovet for matretter. Grunnet dette valgte gruppen å prioritere andre viktige ting som påminnelser, da dette var nødvendig for algoritmen. I oppgaveteksten ble påminnelser listet som «videre utvikling», men gruppen innså raskt hvor essensiell denne funksjonaliteten var for algoritmen, og valgte derfor å prioritere denne funksjonaliteten. Alternativet før påminnelser var at testpersonen måtte manuelt inn i logger å legge inn blodsukker verdien etter måltidet. Dette åpnet for brukerfeil, og gjorde det vanskelig å registrere overlappende måltider, samt at det var svært mye jobb og kronglete for brukeren.

---

## 5.5 Administrativt

Grunnen til at Gantt-diagrammet ble endret, er fordi gruppen fikk et bedre estimat på hvor lang tid enkelte oppgaver ville ta, grunnet erfaringer med tidligere oppgaver. I tillegg jobbet gruppen i flere helger for å starte datainnsamlingen tidligere, som førte til at de fleste arbeidsoppgavene ble forskjøvet frem og arbeidet på tidligere enn originalt planlagt. En annen grunn er at gruppen oppdaget at visse arbeidsoppgaver måtte utføres før andre, og selv om gruppen hadde tatt hensyn til dette, var det enkelte tilfeller som ble oversett. I tillegg hadde ikke gruppen tatt hensyn til at koden skulle testes i det originale Gantt-diagrammet. Ettersom testing er svært viktig for å dokumentere at koden faktisk fungerer, var gruppen nødt til å legge dette inn i planen. Dette medførte visse endringer, som at resultatanalysen ikke ble påbegynt før senere. Dette førte igjen til at testingen av analysen ble utsatt. Ettersom skriving av tester måtte inn i planen, og gruppen fortsatt måtte gjennomføre de arbeidsoppgavene som var listet opp i Gantt-diagrammet, måtte medlemmene finne en løsning på det potensielle tidsproblemet. Gruppen løste det ved at hovedrapporten ble skrevet på i parallell, mens én skrev tester. Ettersom det var satt av to uker til rapportskriving inn mot slutten av prosessen, samt at gruppen planla å være ferdig én uke før fristen, fordelte gruppen denne tiden mer jevnt til andre arbeidsoppgaver som testing av koden.

Et eksempel der en arbeidsoppgave måtte utføres før en annen, som gruppen hadde oversett under utviklingen av det første Gantt-diagrammet, var logger. Funksjonalitet for å logge den satte insulinmengden ble implementert før algoritmen, og dette er fordi uten den loggførte dataen, kan ikke algoritmen gjøre endringer basert på loggført data. Gruppen førte også opp at utviklingen og testingen av resultatanalysen skulle skje før implementasjon av algoritmen. Grunnen til at dette ble byttet om på var at det ikke hadde vært noe å teste uten at algoritmen hadde hatt mulighet til å gjennomføre endringer, og gruppen ønsket å få startet algoritmen så tidlig som mulig, så datainnsamlingen kunne starte så tidlig som mulig. Gruppen oppdaget feilene tidlig, og gjennomførte endringene i Gantt-diagrammet slik at dette skulle være oppdatert.

Etter innspill og råd fra veileder, bestemte gruppen seg for å forskyve og forlenge perioden hovedrapporten skulle bli skrevet på. Gruppen hadde originalt satt av to uker til dette, men konkluderte at det var fornuftig å starte tidligere. Selv om gruppen begynte på rapporten før de hadde fått resultater fra algoritmen, var det mulig å om for eksempel teori, metode, forord og så videre. I tillegg konkluderte gruppen med at dette var arbeid som kunne utføres i parallell med annet arbeid, som ville effektivisere prosessen. Gruppen erfarte at delingen av arbeidsoppgaver i henhold til hovedrapporten, ble unødvendig og mindre effektivt. Medlemmene erfarte at det var bedre å skrive parallelt på rapporten med andre arbeidsoppgaver, og skrive det som kunne skrives til ethvert tidspunkt, enn å ha en rigid struktur på når de ulike delene av rapportarbeidet skulle utføres.

Algoritmen i seg selv tok lenger tid å implementere enn originalt planlagt. Det ble behov for en omfattende omskrivning («rework») av koden før algoritmen kunne implementeres, dette diskuteres i delseksjon 5.7. Denne omskrivningen ble lagt under tiden det tok å implementere algoritmen, da det var denne som krevde omskrivningen. Begge gruppens medlemmer arbeidet på algoritmen, for å sikre at den ble korrekt gjennom parprogrammering. Som nevnt i delseksjon 4.3.1, var det originalt planlagt å finpusse og justere algoritmen. Dette var tenkt for å optimalisere algorit-

---

men. Problemet med dette var at gruppen ikke hadde mulighet til å samle inn nok data, for å undersøke de eventuelle endringene og deres påvirkning. I tillegg måtte gruppen ha testet de ulike variasjonene av algoritmen over lenger tid for å undersøke hvilken som ga et best resultat. For dette vurderte gruppen å lage simulasjoner, men det ble sløffet grunnet at det ikke var satt av tid til det. I tillegg dukket andre viktigere arbeidsoppgaver opp, som testing av koden. Ettersom det er flere ulike aspekter som kan påvirke ens blodsukkerverdier, kunne derfor ikke gruppen, med tiden tilgjengelig, trekke fornuftige konklusjoner om hvilken av de ulike småjusteringene av algoritmen som ga et best resultat, da det også kunne vært at dagsformen i den perioden var bedre. For å kunne teste eventuelle justeringer i algoritmen, som endring av parametere, måtte gruppen ha testet de ulike algoritmeversjonene over en svært lang periode for å anskaffe gode konklusjoner. Gruppen hadde dessverre ikke nok tid til dette, og valgte derfor å bruke lenger tid på å finne gode parametere til algoritmen, og kun bruke den versjonen av algoritmen som beskrevet i delseksjon 3.3.1.3. Dette førte også til mer data på algoritmen, både før og etter at den ble implementert, og som nevnt i delseksjon 3.3.1.3, er det viktig å samle inn nok datapunkter etter den siste avbrytelsen.

Valgene bak å sammenslå og dele opp enkelte arbeidsoppgaver i Gantt-diagrammet, kom av at gruppen så på det som fornuftig. For eksempel ble databasemodelleringsoppgaven endret. Ettersom databasen kunne modelleres ved prosjektets start, men ikke implementeres før den nødvendige kildekoden var skrevet, ble arbeidsoppgaven delt. Implementeringen av databasen skjedde dermed underveis gjennom prosessen da de ulike klassene ble ferdig skrevet.

Som ved tester til koden, var ikke brukertester originalt planlagt inn i Gantt-diagrammet, men gruppen så på det som viktig å få testet brukergrensesnittet. Derfor ble dette lagt til som en arbeidsoppgave.

## 5.6 Samarbeid

Prosjektet har hatt mange ulike arbeidsoppgaver, som for eksempel utvikling og forskning. Gruppen har også arbeidet med å tilegne seg og anvende teori, samt å utvikle et intuitivt brukergrensesnitt som er lett å ta i bruk. Innad i gruppen finnes det ulike styrker og interesseområder for hva medlemmene foretrekker å arbeide med. Ved måten valgte å løse utviklingsoppgavene på, gjennom parprogrammering, har begge fått mulighet til utfylle hverandres kompetanse, og gruppearbeidet har fungert godt. Gjennom parprogrammering har begge studentene kommet med innspill til MVP-en og algoritmen. Parprogrammeringen fungerte godt da gruppen raskere oppdaget eventuelle problemer, enn det medlemmene ville gjort på egenhånd. I tillegg åpnet parprogrammeringen for å utnytte medlemmenes styrker, da de har ulike interessefelt innen utvikling. Et annet positivt aspekt ved parprogrammeringen, er at den førte til at gruppen hadde en felles start på arbeidet hver dag, og at man fikk arbeidet sammen.

Da gruppen fordelte arbeidsoppgavene etter at MVP-en og algoritmen var ferdig implementert, utnyttet gruppen igjen de ulike medlemmenes styrker. Et eksempel på dette er da det ene medlemmet begynte å arbeide på hovedrapporten, mens det andre medlemmet skrev tester til koden. De ulike styrkene i dette tilfellet, er at den ene er god til å produsere nyttig innhold rask til rapporten, og den andre er god til å skrive tester raskt, da begge trivdes med sine respektive oppgaver. Gruppen har følt på

---

et tidspress gjennom hele prosjektet, både grunnet datainnsamlingen, og uventede hendelser som sykdom, var gruppen interessert i å effektivisere disse prosessene så mye som mulig, for å unngå et ennå større tidspress. Det har ikke oppstått konflikter mellom medlemmene under hele prosessen, og dette har bidratt til et godt miljø innad i gruppen. Fordelingen av arbeidsoppgaver har også blitt tildelt basert på hva som var mest fornuftig gjøre, og hvem som var ferdig med sin arbeidsoppgave.

## 5.7 Systemutvikling

Som nevnt i delseksjon 3.10, valgte ikke gruppen å bruke en spesifikk systemutviklingsmetodikk, men heller bruke ulike agile metoder hentet fra ulike systemutviklingsmetodikk. Gruppen opplevde at dette fungerte svært bra, og at tiden som var tilgjengelig alltid ble brukt på en fornuftig måte, som nevnt i delseksjon 3.10. Ved denne prosessen hadde gruppen en svært god oversikt over utviklingen av MVP-en og algoritmen, samt de andre aspektene, som dokumentering, gjennom hele prosjektet. Selv om gruppen erfarte at de sparte tid ved å parprogrammere, opplevde de også at dersom det oppsto for eksempel bugs, at tiden ble brukt litt ineffektivt. Da et slikt tilfelle oppsto, og gruppen innså at det var flere bugs som måtte fikses, og at det ville vært unødvendig og dårlig bruk av tiden at begge medlemmene arbeidet med dette, fordelte gruppen arbeidsoppgavene. Et negativt aspekt ved prosessen var Gantt-diagrammet. Selv om Gantt-diagrammet ble brukt mye og ga en god oversikt over prosessen, holder diagrammet kun oversikt over de overordnede oppgavene. I starten av prosjektet fungerte dette svært godt, da gruppen hadde god oversikt over funksjonaliteten som skulle implementeres i MVP-en, grunnet andre verktøy som wireframe. Problemet oppsto mer mot slutten av prosjektet, da det var mange små oppgaver som måtte gjennomføres. Et eksempel på en slik oppgave er å undersøke at alle møteinnkallinger og møtereferat har lik dato i navnene. Gruppen løste imidlertid dette problemet ved å ha et felles notat der slike arbeidsoppgaver ble notet, i tillegg til bruk av issue-board på github. Issue-board ble brukt mest i forbindelse med utviklingen, mens notatet ble brukt mer for arbeidsoppgaver relatert til dokumentasjonen.

Det var enkelte valg angående teknologien som førte til enkelte utfordringer. Blant annet måtte gruppen ta i bruk SQLite. Ettersom dette er en mindre versjon av SQL, var det ikke mulig å modellere databasen med like mye frihet. I tillegg til at spørringen mot databasen ikke kunne være like spesifikke, og de var begrensede. En annen utfordring som oppsto var distribusjonen av MVP-en. Ettersom gruppen utvikler MVP-en for en iPhone, da det er dette testpersonen har, var det diverse sertifikater som måtte på plass for å få MVP-en på den mobile enheten. Dette bydde på ulike utfordringer ved distribusjonen av MVP-en. Det var mye mer arbeid som måtte til for å få alle disse sertifikatene på plass slik at man kunne kjøre MVP-en på den mobile enheten.

Det var også behov for en omfattende omskriving («rework») av koden før algoritmen skulle bli implementert. Denne omskrivningen var nødvendig for å opprettholde høy cohesjon og lav coupling. Modellene som representerte logger, påminnesler, ol. ble delt i to slik at det var en, DAO, som representerte hvordan objektet så ut i databasen, og en som hadde alle informasjon om hvilke objekter den var koblet til samt hjelpelogikk for algoritmen. Dette sikret høy cohesjon da ansvarsområdet til de separerte klassene var tydeligere og mer snevert. Samtidig gjorde hjelpelogikken at det ble mindre coupling mellom algoritmen og modellene. Dette da modellene hadde



---

informasjonen, gjennom hjelpelogikken, algoritmen trengte, så den ikke trengte å hente ut all informasjonen fra modellen å finne ut av dette selv. Dette ansvaret faller innenfor modellens ansvarsområdet og sikrer høyere cohesjon. Videre ble det også oppdaget at både frontend og algoritmen brukte database-klasser direkte som gjorde at de selv måtte sikre at koblinger mellom objekter ble fulgt etter reglene. For eksempel så skal det ikke eksistere logger med en matvare som er slettet. Dette kunne algoritmen og frontend bryte om koderen glemte å legge inn dette. Dette blir repetisjon av kode, og om reglene endres så vil man måtte finne alle steder hvor dette skjer. Først ble deler av logikken flyttet til database metodene, men det var ikke i alle tilfeller man ville ha disse reglene. Det var heller ikke database-klassenes ansvar å sikre at reglene mellom objektene ble opprettholdt, deres ansvar var å ha sikker kommunikasjon med databasen. For å oppnå høy cohesjon ble to lag til laget: «Service» og «Repository». Service-laget ble et grensesnitt for algoritmen og frontend. Dette tok seg av alle reglene som gjaldt mellom objekter av forskjellige typer, det ble laget en service-klasse per modell som ble brukt i frontend og algoritmen. Service-laget brukte repository-laget for å hente og sende objekter. Repository-laget hadde som oppgave å oversette en modell til dens respektive DAO-modell. Deretter ble oppgaven sendt videre til database-laget (også kalt «Persistence»-laget). Database-laget hadde nå som eneste ansvar å sikre sikker kommunikasjon mellom koden og databasen. Denne store arkitecturendringen sikret høy cohesjon og lav coupling, slik at koden var lettere å jobbe med videre, samt hadde bedre ytelse.

En fordel ved valg av frontend-rammeverk var at utviklingen av MVP-en gikk relativt raskt, da man kunne bruke ferdigkomponenter for designet. En annen fordel ved teknologivalg var at C# kodespråket minnet mye om tidligere kjente språk, som Java. Dette gjorde at gruppen kom raskt i gang med utviklingen, og at tiden ikke ble brukt på å lære seg et nytt språk.

Gruppen opplevde at den planlagte fremgangsmetoden, beskrevet i vedlegg A, fungerte godt. Gruppen hadde den motivasjonen og erfaringen som var nødvendig for å lage en problemstilling for oppgaven. Dette var utfordrende da oppgaven i seg selv er svært omfattende, og det var vanskelig å verken lage oppgaven for snever eller for stor. Ulike fordeler og ulemper ved fremgangsmetoden har blitt nevnt i delseksjon 3.1. Det som gikk bra på grunn av fremgangsmetoden var at den ga en overordnet oversikt over hvordan gruppen skulle gå frem for å svare på forskningsspørsmålene. Noe som gikk dårlig grunnet fremgangsmetoden, var tidspresset rundt case study-en. Dette har allerede blitt diskutert i større detalj tidligere. Grunnet valget av case study som strategi, ville tiden bli kritisk, men dette var gruppen klar over, og de gjorde sitt beste for å motvirke dette. Brukertestene, var noe som gikk svært bra ved fremgangsmetoden. Gruppen fikk god innsikt i designet og hva som brude endres.

Uavhengig av prosess, fremgangsmåte og teknologi, fungerte samarbeidet, som nevnt i delseksjon 5.6, svært godt. Gruppen tror dette kommer av at medlemmene hadde en god innstilling til oppgaven, og at begge var motiverte for arbeidet som måtte til for å gjennomføre oppgaven, samt god kommunikasjon innad i gruppen. Det som gikk mindre bra, var tidspresset angående datainnsamlingen. Som vist i Gantt-diagrammet, var det fysisk ikke mulig for gruppen å starte prosessen tidligere. Ettersom det ikke var mulig å utvide denne perioden, var det heller ikke mulig å gjennomføre endringer i algoritmen, og få testet disse ordentlig.

---

## 5.8 Feilkilder

### 5.8.1 Bias

Ettersom testpersonen også er oppgavestiller og utvikler, var gruppen klar over at vedkommende kunne ha et bias. Dette er en av grunnene til at gruppen valgte å gjennomføre brukertester for å undersøke hvor intuitivt designet var. Det var imidlertid ikke en stor fare for at testpersonen kunne påvirke datainnsamlingen i noen stor grad, da testpersonen må føre inn tallverdier og hva som ble konsumert til et måltid. Testpersonen har vært ærlig med dataene som har blitt ført inn i applikasjonen for å gi algoritmen den korrekte dataen, slik at den kan forbedre seg.

### 5.8.2 HbA1c-verdi

HbA1c-verdien til testpersonen har blitt kontinuerlig hentet ut gjennom prosessen, med en ukes intervall gjennom testpersonen sin sensor applikasjon, LibreLink. Den har altså ikke blitt målt av helsepersonell. Dette kan være en potensiell feilkilde, da HbA1c-verdien hentet fra LibreLink kun er et estimat av denne verdien. Ettersom HbA1c-verdien har blitt hentet ut på lik måte, antok gruppen at LibreLink sine estimater var gode nok til å gi et godt innblikk i hvordan HbA1c-verdien har endret seg gjennom prosessen, selv om det kun er estimater. Grunnen til at gruppen ikke målte HbA1c-verdien hos helsepersonell, er at gruppen ville ha en kontinuerlig oppdatering av HbA1c-verdien og ikke kun en måling før, under og etter prosessen. Det var heller ikke praktisk for testpersonen å teste seg hos helsepersonell like ofte som gruppen ønsket oppdateringer av HbA1c-verdien.

### 5.8.3 Trening

Selv om gruppen har forsøkt å minimere feilkilder under datainnsamlingen, vil det alltid være noen som oppstår. En av disse feilkildene er treningsfrekvensen til testpersonen. Både før og under testperioden hadde ikke testpersonen en jevn treningsfrekvens. Som nevnt i delseksjon 2.1 vil trening påvirke insulinsensitiviteten. Der som man ikke har faste rutiner for trening vil denne forskjellen være større. Ettersom testpersonen ikke hadde en jevn treningsplan, ble derfor visse insulindoser justert. MVP-en har ikke funksjonaliteten «events» som hadde tatt hånd om å separere data påvirket av trening fra annen data, så denne blandingen av data kan også ha spilt inn i noen tilfeller. Ettersom trening er viktig for ens helse, var det ikke aktuelt at testpersonen ikke skulle få lov til å trene gjennom testperioden.

### 5.8.4 Mangel på data

En annen mulig feilkilde er at det ikke er nok data for å kunne trekke riktige konklusjoner. Ettersom datainnsamlingen ikke begynte før 8. mars, var det ikke mulig å samle inn like mye data som gruppen hadde ønsket. Selv om det var visse elementer det ble samlet inn en god del data på, som matvaren «brød» og de ulike dagsprofilene, var det flere matvarer det ikke ble samlet inn nok data på for at algoritmen kunne gjennomføre endringer. Dette kan ha medført at algoritmen ikke fikk nok tid til å tilpasse seg dataene, og at trendene man kan observere ikke er ferdigutviklet

---

eller representerer at algoritmen forsøker å justere seg.

#### **5.8.5 «Insulin-til-glukose»-forholdet**

Som nevnt i tidligere delseksjon 5.2.1.2, skapte den kontinuerlige justeringen av denne verdien store problemer. Gruppen er under oppfatning at det kunne blitt et bedre resultat, dersom denne verdien hadde vært fast. Det er ikke sikkert om algoritmen hadde lært bedre og kunne forbedret reguleringen, selv om denne hadde vært en fast verdi. Gruppen er derimot meget sikker på at algoritmen ikke hadde blitt like feil som den ble på slutten av testperioden.

---

## 6 Konklusjon og videre arbeid

Basert på resultatene i seksjon 4 og diskusjonen i seksjon 5 kan man trekke en konklusjon på de presenterte forskningsspørsmålene: «*Hvor presist kan en algoritme basert på tidligere loggført data beregne en korrekt insulinmengde for å oppnå ønsket blodsukker verdi?*» og «*Hvor mye tid kan sluttbruker spare i uken ved å ta i bruk MVP-en?*».

### 6.1 Konklusjon

Algoritmen og MVP-en utviklet i oppgaven kan benyttes for å estimere insulinmengder for diabetikere. Basert på de preliminare resultatene fremstilt i ulike figurer i seksjon 4, anslår gruppen at algoritmens presisjon ikke er like god som ønsket. Svaret på det første forskningsspørsmålet har gruppen kommet frem til, gjennom analyser av figurene, er en presisjon som holder til en HbA1c-verdi på 7,5 prosent. Med flere iterasjoner av algoritmen, og justeringen av verdien «insulin-til-glukose»-forholdet til en fast verdi, som nevnt i diskusjonen 5.2.1.2, vil denne presisjonen mest sannsynlig øke, og gi et bedre resultat, altså en lavere HbA1c-verdi. Dette krever derimot en ny testperiode for å bekrefte hypotesen. Om presisjonen på 7,5 prosent er en akseptabel presisjon er usikkert. Dersom enkelte diabetikere er bedre regulert uten denne algoritmen, ville det ikke nødvendigvis vært nyttig for disse å ta i bruk MVP-en i håp om å bli bedre regulert. Basert på de preliminare resultatene og presisjonen til algoritmen, kan dette derimot være et godt verktøy for de som har en lignende eller lavere presisjon. Det kan også være et godt verktøy for de som ønsker å bruke mindre tid på reguleringen av blodsukkeret, og kanskje få en forbedret HbA1c-verdi.

Fra de preliminare resultatene om tidsbruk er det også mulig å konkludere på det andre forskningsspørsmålet. Basert på de preliminare dataene, kan en sluttbruker av MVP-en spare gjennomsnittlig omtrent 40% av tiden brukt på å regulere blodsukkeret i uken. Dette er en betraktelig reduksjon, selv om det ikke oppnådde målet, som var ambisiøst. Dette resultatet kan bidra til at diabetes blir en mindre tidkrevende sykdom.

### 6.2 Videre arbeid

Opgaven har et potensial, og det eksisterer flere aspekter som kan medføre en mer presis algoritme. Ved videre arbeid kunne man implementert funksjonaliteten «events» som beskrevet i visjonsdokumentet. Denne funksjonaliteten tar hensyn til ulike hverdagshendelser, for eksempel trening og sykdom, som krever en justert insulinmengde. Det ville vært svært interessant å se hvordan algoritmen utviklet seg ved å ta hensyn til slike variabler, og hvilken presisjon den ville hatt.

I tillegg kunne man koblet opp MVP-en mot sensorer for måling av blodsukkerverdier, for å få en kontinuerlig oppdatering av blodsukkerets utvikling. Ved å gjøre dette hadde man hatt mye mer data som kunne ført til at kategoriseringen av overlappende måltider hadde vært detaljert, og man kunne gjennomført mer informerte valg og fordelinger.

Man burde også implementere en fast verdi for «insulin-til-glukose»-forholdet, slik

---

som beskrevet i delseksjon 5.2.1.2, for å undersøke nærmere om hypotesene om et bedre regulert blodsukker er sanne.

En bedre løsning kan være å la bruker spesifisere hvor mye insulin de vanligvis setter til matvaren de legger inn. På denne måten kan man justere skalaren til å stemme overens med denne verdien med en gang. Utgangspunktet blir da ikke de teoretiske verdiene, men heller hva bruker setter fra før. Fram til algoritmen gjør noen endringer vil dette teoretisk kunne oppnå akkurat samme regulering som bruker har uten MVP-en, da de bruker like verdier. Derimot vil de bruke mindre tid på reguleringen.

Annet arbeid kan også være å utføre et større eksperiment med flere testpersoner, for å undersøke hvordan algoritmen reagerer og justerer seg til andre individer. I tillegg burde testperioden forlenges for å samle inn den nødvendige dataen på de ulike aspektene som trening, diverse matvarer og dagsprofiler, slik at algoritmen kan foreta flere iterasjoner på disse.

Dette prosjektet kan bli ansett som et godt fundament for videre arbeid og forskning, for å oppnå et ennå bedre resultat. Arbeidet så langt viser et potensiale fra de preliminære dataene, og gruppen mener at oppgaven er verdt å utforske mer.

---

## 7 Samfunnspåvirkning

### 7.1 Etikk

Det finnes ulike profesjonsetiske problemstillinger i forhold til gruppens gjennomføring av oppgaven og hva gruppen har vurdert underveis i prosessen. Ved en profesjonsetisk problemstilling menes problemstillinger knyttet til et yrke hvor utøverne skal ha og anvende en viss ekspertise rundt emnet, feltet eller vitenskapen. Dette gjelder spesielt yrker der det er krav om lang utdanning og formelle kvalifikasjoner [46]. Det er svært viktig at man som utvikler tar gode etiske avgjørelser, og at man er klar over hva slags ansvar man har ovenfor hva man utvikler. Man har som utvikler en mulighet til å kontrollere hva slags teknologi som blir skapt.

Det er viktig å være klar over at selv om man kan utvikle ny teknologi, er det ikke alltid man bør. Man bør alltid undersøke alle mulige konsekvenser et utviklet produkt kan ha. I MVP-ens tilfelle, er dette ment til å være et hjelpemiddel for diabetikere. Presisjonen til MVP-en måles fra HbA1c-verdien, da dette er en praksis helsepersonell bruker. I det tilfellet, MVP-en skulle gi en diabetiker inntrykk av at blodsukkeret deres er dårlig regulert, kan dette by på uventede konsekvenser. Det kan for eksempel gå ut over ens personlige økonomi, og helse.

Et annet viktig aspekt er personvern. Det har blitt gjennomført to brukertester, der begge har blitt anonymisert og gitt tallkoder. Brukertestene kan dermed ikke spores tilbake til personene. Ettersom testpersonen utfører forskning på seg selv, og dette åpner for at testpersonen kan ta avgjørelser selv, og gjøre hva de vil. Testpersonen har ikke blitt anonymisert, da vedkommende følte det var viktig å få frem at de var en del av gruppen som utviklet MVP-en. Til tross for at testpersonen selv bestemmer hva slags forskning vedkommende kan utføre på seg selv, utvikler gruppen et verktøy som er ment som et helseverktøy. Gruppen har derfor vært nøye med å tenke gjennom ulike konsekvenser dette kan ha, både for testpersonen og andre, å ta i bruk MVP-en.

Det eksisterer flere etikkoder for ingeniører. Blant disse, beskrevet av NSPE i «*Code of Ethics for Engineers*», er det en av flere problemstillinger gruppen arbeidet med, og dette er «*Hold paramount the safety, health and welfare of the public*». Dette går ut på å ta nøye hensyn til sikkerheten, helsen og velferden til mennesker. Ettersom MVP-en er et medisinsk hjelpemiddel, er det svært viktig at algoritmen er nøyaktig og korrekt. Dette er fordi den direkte påvirker diabetikere sin helse gjennom blodsukkerreguleringen. Grunnet dette har gruppen vært nøye med å understreke at dette er en MVP, og ikke en fullverdig applikasjon. I tillegg undersøker gruppen mer i detalj om algoritmen, og MVP-en i seg selv, er verdt å videreutvikle, ut ifra de preliminare dataene. Gruppen har også skrevet tester for backend-koden, der algoritmen er skrevet, i tillegg til mye unntakshåndtering, for å sikre kodens integritet. Dette er beskrevet i mer detalj i delseksjon 3.2.6.

NITO har også egne etiske retningslinjer. Blant disse finner man blant annet at «NITOs medlemmer og ansatte skal vedkjenne seg sitt faglige ansvar og utføre sitt arbeid i henhold til anerkjente kvalitetsnormer» [47]. Gruppen hadde god kjennskap til og gode kunnskaper om diabetes type 1 ved starten av prosjektet, grunnet egne erfaringer ved sykdommen. Det var viktig for gruppen å ha en god forståelse av diabetes, og hvordan algoritmen skulle utarbeides. Gruppemedlemmene fikk kontinuerlig diskutert under utviklingsprosessen hvordan algoritmen skulle utformes, og

---

dette medførte at gruppen fikk delt sin kunnskap, og sørget for å lage et produkt basert på riktig kunnskap. Kvalitetsnormene gruppen tok hensyn til var da allerede kjente insulinberegnings metoder, samt diabetes type 1 i sin helhet. Ettersom gruppen utarbeidet en MVP, konkluderte gruppen at det kun skulle tas utgangspunkt i de preliminære dataene for å undersøke om ideen og algoritmen er noe som burde videreutvikles til et godkjent medisinsk verktøy. Det finnes andre ulike kvalitetsnormer for medisinske verktøy, som gruppen i mindre grad tok hensyn til grunnet tid, og at det er en MVP som blir utviklet.

Andre profesjonsetiske problemstillinger som var aktuelle omhandlet NITO sin etiske retningslinje om at man «skal vise respekt for samspillet mellom teknologi og menneskelige verdier, samt bidra til åpenhet om konsekvenser av teknologiske løsninger.» [47]. Her har gruppen kontinuerlig gjennom prosessen reflektert om MVP-en har noen konsekvenser, og om tjenesten den leverer. Gruppen har konkludert med at dersom det er noen store mangler med koden, som ikke blir plukket opp av tester, unntakshåndtering og uttesting, kan MVP-en være svært farlig å ta i bruk. Dette er fordi det er vært få enheter med insulin som skal til før noe kan gå galt. For eksempel kan man oppleve en føling dersom man gir for mye insulin, se delseksjon 2.1. Grunnet dette har gruppen lagt inn en sikkerhetsmekanisme i koden for å unngå dette. Dette knytter også opp imot den førstnevnte profesjonsetiske problemstillingen i denne delseksjonen.

Tekna trekker frem to poeng gruppen har vurdert og diskutert gjennom prosessen, nemlig om MVP-en «har uheldige konsekvenser for individer, miljø eller samfunn» [48] og om den «tåler offentlig oppmerksomhet» [48]. Det førstnevnte er noe som vil bli diskutert i mer detalj videre i seksjon 7. Sistnevnte er noe gruppen har valgt å nedprioritere grunnet tid og det faktum at det kun blir utviklet en MVP og ikke en fullstendig applikasjon, som nevnt tidligere. Selv om det ikke har vært en prioritet, har gruppen arbeidet for at det skal være lett å videreføre MVP-en og gjøre den klar for offentlig oppmerksomhet. En annen grunn til at denne problemstillingen ikke har blitt prioritert, er grunnet alt det juridiske og offentlige dokumenter som må på plass før en slik applikasjon kan offentliggjøres. Det er derfor viktig at dersom MVP-en videreutvikles til en fullverdig applikasjon, må den gjennom en større testprosess med en større testgruppe enn det som har blitt gjort under denne prosessen.

En gjennomgående problemstilling har vært nøyaktigheten av algoritmen og insulinberegningen. Som beskrevet i tidligere delseksjon 2.1, kan det ha fatale konsekvenser om insulinberegningen blir veldig ukorrekt. Gruppen har vært nøye med å ha kontinuerlig code review gjennom parprogrammering, testing av koden, og testing av algoritmen og MVP-en gjennom prosessen, for å fange opp eventuelle feil. I tillegg til at det er blitt implementert funksjonalitet for å endre den gitte insulinmengden etter brukers behov og vaner.

Gruppen har gjennom prosessen forsøkt å holde seg til, reflektere og vurdere NITOs og Teknas etiske retningslinjer, og hvordan disse passer med gruppens oppgave. Ettersom gruppen hadde god erfaring med diabetes og hva slags konsekvenser denne sykdommen kan ha, var begge veldig oppmerksomme på effekten av å ikke ta hensyn til de etiske problemstillingene rundt oppgaven.

---

## 7.2 Vitenskap

### 7.2.1 Prosess

Rundt gruppens prosess finner man både styrker og svakheter knyttet til vitenskape-lighet. En av styrkene rundt prosessen var den grunnleggende forståelsen, de gode kunnskapene, og erfaringene gruppen hadde rundt diabetes type 1. Dette ga grunnla-get til å analysere dataen korrekt, utarbeide en god algoritme, samt utarbeide en MVP for innsamling av nødvendig data. I tillegg til gruppens eksisterende kunnskap, leste medlemmene relevante forskningsartikler [33] [31] [14] [7] [10] innenfor flere fag-felt for å bekrefte at kunnskapen som er innhentet og systematisert i samsvar med grunnleggende krav som den vitenskapelige kulturen anerkjenner [49]. Gruppen har gått systematisk frem for å bygge ny kunnskap og forstå den allerede eksisterende kunnskapen bedre. I tillegg har gruppen vært nøye med å håndtere kunnskapen på en systematisk og ærlig måte.

I følge boken «*Kvalitative metoder i medisinsk forskning*» har man følgende grun-nbetingelser for vitenskapelig kunnskap: relevans, validitet og refleksivitet [49]. Rel-evans omhandler hva kunnskapen blir brukt til. Hva forskeren egentlig har funnet ut av er gyldighet eller validitet. Refleksivitet handler om hvordan prosessen har påvirket funn og konklusjoner.

Algoritmen utviklet under prosjektet oppfylder relevans kravet. Dette er fordi det er en ny algoritme og en ny måte å beregne insulinmengde på, og bidrar dermed med noe nytt innenfor fagfeltet.

Validitet omhandler tre viktige spørsmål: intern validitet, ekstern validitet og repeter-barhet [49]. For intern validitet har gruppen tatt i bruk relevante begreper og verktøy for å finne den relevante kunnskapen. For eksempel har gruppen tatt i bruk kunnskap angående diabetes og statistikk for å utvikle MVP-en. Her var gruppen nøye med å bruke anerkjente formler for statistikken. Det var viktig at MVP-en faktisk samlet inn den korrekte dataen, og at gruppen visste hva de skulle gjøre med denne. Dette er blant annet hvorfor gruppen valgte å utvikle MVP-en, da eksisterende data ikke ga gruppen den dataen som var nødvendig for algoritmen. I MVP-en er det også im-plementert sikkerhetsmekanismer og unntakshåndtering for å forsikre at dataene er pålitelige. Ekstern validitet, eller overførbarhet, handler om hvordan funnene i opp-gaven kan gjøres gjeldene utover den konteksten den har blitt kartlagt i [49]. For eksempel vil algoritmen ikke kun fungere for én person. Ettersom algoritmen bruker dataen brukeren av MVP-en legger inn, vil den dermed også tilpasse seg hver enkelt person. Repeterbarhet handler om å kunne gjenta forsøket og ende opp med like resultater hver gang. Ettersom algoritmen justerer seg basert på hver enkelt individ sitt blodsukker, vil den alltid gi ulikt svar for andre testpersoner enn testpersonen under prosjektet. Grunnet tid hadde heller ikke gruppen mulighet til å nullstille al-goritmen og repetere eksperimentet, for å undersøke om algoritmen ville finne de samme verdiene etter en gitt tid. I tillegg ville det vært svært komplisert å gjenta eksperimentet, da det ikke kun omhandler et likt tidsrom, men også hva slags mat testpersonen spiser, hvor mye mat som konsumeres, sykdom og trening. Dersom man skulle gjennomført eksperimentet på nytt, burde det gjennomføres med flere iterasjoner for å undersøke om algoritmen finner omtrentlig de samme verdiene hver gang.

Refleksivitetskravet oppfylles ved at gruppen kontinuerlig har vurdert fremgangs-



---

måten og konklusjoner. Ved valg av teknologi og metode vurderte gruppen flere valgmuligheter, og drøftet positive og negative aspekter ved de ulike valgene. Gruppen har også beskrevet i detalj de ulike kontekstuelle forutsetningene for å tolke problemstillingene. Ved å teste deler av koden, har gruppen også søkt konfrontasjoner med egne posisjoner, for å undersøke feil. I tillegg har gruppen vært nøye med å beholde integriteten i resultatene, for eksempel ved å forkaste korrupt data selv om dette medførte en mindre datainnsamling.

### **7.2.2 Resultater**

Det er en vitenskapelig relevans for algoritmen, da dens resultater bidrar til ny informasjon innenfor et eksisterende fagfelt. Fra resultatene kan man for eksempel videreutvikle algoritmen, og forbedre denne, og på denne måten drive fagfeltet videre.

Resultatene fra algoritmen er valide. For det første er det brukt relevant kunnskap innenfor medisin (diabetes type 1) og statistikk for å undersøke resultatene og konkludere ut ifra disse. Dette oppfylder spørsmålet om intern validitet. Det er blitt brukt statistikk for analysen av resultatene, og valget av ITS analyse har påvirket resultatene og hvordan disse blir tolket. For ekstern validitet, har det blitt diskutert i delseksjon 7.2.1 om hvorvidt det er mulig med overførbarhet. Fra denne diskusjonen ble det konkludert at det vil bli vanskelig å få nøyaktig de samme resultatene, da algoritmen tilpasser seg hver enkelt person. Likevel er gruppen positive til å oppnå lignende resultater. Dette for eksempel innen tidsbruk og reduksjonen av denne.

Proessen har påvirket resultatene i den grad av at testperioden har bestått av både sykdom, uventet fysisk aktivitet, og uregelmessig trening. I tillegg har matvarer som kun har blitt spist noen få ganger, ikke fått gjennomgått en iterasjon med algoritmen, og dermed har den ikke kunnet forbedre disse. Valget om å gjennomføre en case study, og ikke simulasjoner, har også påvirket resultatet. Ved en case study, har algoritmen fått operert i det mest naturlige miljøet. I tillegg har dette ført til at det har vært mulig å estimere tidsbruk, både før og under MVP-en. Dette ville vært vanskelig å simulere. Ved å ha en testperiode over lenger tid vil dette ha påvirket resultatene, og det er mulig gruppen ville fått en annen presisjon på algoritmen gjennom simulasjon. En ulempe ved å ikke ha simulasjoner, er at man kunne justert mer på algoritmen, og gjennomført flere simulasjoner for å finne parametere som ga de mest optimale resultatene.

### **7.3 Bærekraft**

Bærekraft blir definert som et begrep for «å karakterisere økonomiske, sosiale, institusjonelle og miljømessige sider ved menneskelige samfunn» [50]. Bærekraftig utvikling beskriver en «utvikling som tilfredsstillende behovene til mennesker som lever i dag, uten å ødelegge fremtidige generasjoners muligheter for å tilfredsstillende sine behov» [51]. MVP-en er i stor grad bærekraftig, spesielt da den har et stort potensiale til å gagne samfunnet, spesielt med tanke på diabetikere som har behov for en slik løsning som oppgaven fremstiller. I tillegg vil den ikke bruke mye ressurser, se delseksjon 7.3.1 og den vil dermed ikke ødelegge for fremtidige generasjoner, samtidig som den vil løse et problem for dagens diabetikere. Ved å forebygge alvorlige helsekonsekvenser med et bedre regulert blodsukker, gjennom MVP-en, vil dette gagne samfunnet med at, for eksempel, sykehus får mindre innleggelseser. Det vil

---

også gagne enkeltindivider både økonomisk og helsemessig, se delseksjoner 7.3.3 og 7.3.2.

Et samfunnsresultat som prosjektet er med på å fremme er altså å utligne forskjeller for diabetikere som bruker mye tid på å beregne korrekt insulinmengde, samt diabetikere som lider av et dårlig regulert blodsukker. MVP-en, som vist i delseksjon 4.2.1.1, reduserte tiden brukt på å regulere blodsukkeret med omtrent 40%. Dette gjør at diabetikere kan bruke denne tiden på andre aktiviteter, og ha en mer tilnærmet hverdag som personer uten diabetes. Ved å ha et bedre regulert blodsukker, kan dette også bety at diabetikere kan tilbringe mindre tid hos helsepersonell. Dette kan medføre at de som sliter mest får lettere tilgang på den profesjonell hjelpen de trenger.

### 7.3.1 Miljø

MVP-en tar i bruk en lokal database for lagringen av data. Den tar ikke i bruk en sky-løsning, som betyr at det ikke er et behov for eksterne servere for å drifte applikasjonen. Dette gjør at klimaavtrykket MVP-en etterlater seg er svært lite. I forskning-artikkelen «Lifecycle energy assessment of mobile applications» [52] blir det beskrevet at den mobile Android applikasjonen, «Mail K9», sine utslipp består av 125.213 kWh, og dette tilsvarer omtrent 65 kg CO<sub>2</sub>. Dette er basert på antagelser som blant annet at utviklere konsumerer 1,4 kW per time i måneden, én million nedlastinger av applikasjonen, og at Mail K9 applikasjonen konsumerer gjennomsnittlig 0,3 W per time. Applikasjonen sine utslipp blir beskrevet i artikkelen som relativt små. Man kan sammenligne disse resultatene med MVP-en. Beregningene beskrevet i vedlegg C, kan man observere at utslippene til oppgavens løsning er på  $21\text{kg CO}_2\text{-ekvivalenter/aar}$  om alle nordmenn med diabetes type 1 brukte MVP-en, til sammenligning brukte gjennomsnittlig en innbygger i Norge rundt 7,03tonn CO<sub>2</sub> i 2018 [53]. Utslippene fra bruk av MVP-en er derfor meget små.

Alternativet til MVP-en er enten å ta i bruk hoderegning eller bruke en kalkulator. Hoderegning krever kun tid av personen som utfører beregningene, mens en kalkulator bruker et batteri, enten en mobil enhet sitt eller et eget et for en kalkulator. Man kan argumentere for at MVP-en ikke utgjør en spesielt stor forskjell fra disse metodene, og at utslippene ikke vil variere i stor grad basert på hva man velger å benytte seg av for insulinberegning. MVP-en påvirker altså ikke miljøet i en stor grad, spesielt ikke i tilstanden ved prosjektets slutt.

### 7.3.2 Samfunn og helse

MVP-en og algoritmen utviklet under dette prosjektet er i hovedsak et medisinsk verktøy, med tendenser mot et velferdstilbud da det alt eksisterer andre valg for å beregne insulinmengder. Dette er for eksempel insulinpumper, som har deler av funksjonaliteten implementert i MVP-en. Målet med MVP-en er å simplifisere insulinberegningen, samt beregne en mer korrekt insulinmengde. Det er viktig at diabetikere føler seg komfortable med hva slags teknologi de utnytter for å måle blodsukkerverdier og hva de bruker for å sette insulin. Selv om en insulinpumpe har deler av funksjonaliteten til MVP-en, er det ikke alle diabetikere som ønsker å ta i bruk en insulinpumpe. Det finnes ulike grunner til dette, og en av de kan være at de ikke er komfortable med å ha et medisinsk verktøy festet på kroppen til enhver tid, eller at de ikke har

---

økonomisk råd til det. MVP-en er kun en mobilapplikasjon, og er lett tilgjengelig, har mer funksjonalitet og er ikke påtrengende på samme måte som for eksempel en insulinpumpe.

MVP-en vil påvirke en diabetiker sin helse. Dette gjennom reguleringen av blodsukker. Slik som MVP-en var ved prosjektets slutt, tilsier analysen av de prelinære resultatene, at en diabetiker vil bli dårligere regulert. Gruppen tror imidlertid at de har identifisert problemet med algoritmen, og med de rette justeringene, tror de at algoritmen enten vil få en bedre presisjon, eller ha samme presisjon som diabetikeren hadde før algoritmen. Dersom MVP-en ikke forbedrer reguleringen av blodsukkeret, kan dette påvirke en diabetiker sin helse negativt. Dette vil igjen føre til en belastning på samfunnet, gjennom helsetjenestene. I tillegg kan det være en økonomisk belastning på personen det gjelder, se delseksjon 7.3.3. Derimot, om algoritmen enten får de nødvendige justeringene eller blir fjernet fra MVP-en, kan MVP-en fungere som originalt planlagt. Dette innebærer da at diabetikere blir lignende regulert og sparer tid, som vil føre til at positiv samfunnsutvikling, og nå et samfunnsresultat om å utligne forskjeller i en helsemessig situasjon.

### **7.3.2.1 FNs bærekraftsmål**

FNs bærekraftsmål går ut på at verden skal jobbe sammen for å utrydde fattigdom, bekjempe ulikhet, og stanse klimaendringer innen 2030. Av FNs 17 bærekraftsmål bidrar dette prosjektet til mål nummer tre, «God helse og livskvalitet». Dette målet går ut på å «sikre god helse og fremme livskvalitet for alle, uansett alder» [54]. Ved dette målet er det spesielt undermål 3.4 som resonerer med dette prosjektet. Dette undermålet tar for seg å «reducere prematur dødelighet forårsaket av ikke-smittsomme sykdommer med en tredel gjennom forebygging og behandling, og fremme (...) livskvalitet innen 2030» [54]. Etersom diabetes kan ha alvorlige konsekvenser dersom den ikke blir håndtert, kan dette føre til tidlig død i de verste tilfellene. I tillegg er det en tidskrevende sykdom, som også er erfart av testpersonen. Ved å senke mengden tid brukt på diabetes, i tillegg til å enten forbedre glukosenivåene eller la disse være upåvirket, kan man anta at dette vil fremme livskvalitet hos de fleste som sliter med dette. Grunnen bak denne antagelsen er den positive opplevelsen testpersonen hadde med MVP-en. Prosjektet, med nødvendige justeringer, kan bidra både til forhindre dødelighet, samt fremme livskvalitet for diabetikere.

### **7.3.3 Økonomi**

Det er viktig å sette oppgaven inn i et økonomisk perspektiv. Dersom MVP-en skal videreutvikles til en faktisk applikasjon som skal bli brukt av flere diabetikere, kan dette påvirke hvert av disse enkelt individenes økonomi. Dette er fordi dersom algoritmen ikke skulle klare å foreslå riktige insulinmengder, enten som følge av dårlig oppfølging av sluttbruker eller at algoritmen ikke skulle klare å tilpasse seg denne personen, kan dette medføre store konsekvenser i form av senskader eller følinger for enkelt personen. Dette kan føre til at diabetikeren vil forsøke å ordne problemet selv, eller at de oppsøker ellers unødvendig helsehjelp. Som diabetiker får man insulin og annet nødvendig utstyr på resept, men å dra til for eksempel fastlegen har en fast pris uavhengig av sykdom. I de tilfellene applikasjonen skulle medført senskader, kan disse bli dyre. På denne måten kan MVP-en påvirke ens økonomi negativt. På

---

den andre siden, dersom algoritmen gjør som forventet, og gir et bedre regulert blod-sukker, kan det bety mindre oppfølging av helsepersonell, som kan spare en for slike økonomiske belastninger. Disse økonomiske belastningene vil avhenge av hvor i verden man befinner seg. Land som Norge har et godt utviklet helsesystem som gjør at det ikke utgjør en økonomisk forskjell grunnet frikort ordningen [55].

En annen side som er viktig å sette inn i et økonomisk perspektiv er om MVP-en videreutvikles til en fullverdig applikasjon, og deretter skal bli solgt kommersielt. Ved at applikasjonen selges åpner dette for at noen skal tjene penger på den. Dette åpner igjen for en markedsanalyse, og hvem applikasjonen kunne blitt solgt til. Dette kan enten være et større firma, eller at enkelt personer selv må betale for løsningen. Man må også vurdere hvordan og om man kan selge produktet med løfter om å forbedre blodsukkeret til diabetikere. Dette er analyser og beslutninger som gruppen selv ikke vil ta stilling til, da MVP-en kun er en MVP, og ikke en fullverdig applikasjon. Dersom MVP-en videreutvikles, er det disse utviklerne som må ta stilling til dette, men det er viktig at gruppen er klar over hva slags implikasjoner dette kan ha. Som de første utviklerne av MVP-en er det viktig at gruppen er klar over ansvaret av å utvikle en helserelatert MVP, og hva slags konsekvenser dette kan ha. Oppdragsgiver har imidlertid bestemt at dette er et produkt som sluttbruker ikke skal måtte betale for.

---

## Referanser

- [1] N. helseinformatikk. 'Glukose - blodsukker'. (2019), [Online]. Available: <https://nhi.no/sykdommer/hormoner-og-naring/diabetes-generelt/glukose/> (visited on 4th Mar. 2022).
- [2] G. G. L. Lamport, *Statistikk for universiteter og høyskoler*, 4th ed. Universitetsforlaget, 2018.
- [3] Wikipedia. 'Code review'. (2021), [Online]. Available: [https://en.wikipedia.org/wiki/Code\\_review](https://en.wikipedia.org/wiki/Code_review) (visited on 18th May 2022).
- [4] Microsoft. 'The model-view-viewmodel pattern'. (2021), [Online]. Available: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm> (visited on 18th May 2022).
- [5] J. Eder, G. Kappel and M. Schrefl, 'Coupling and cohesion in object-oriented systems', Citeseer, Tech. Rep., 1994.
- [6] H. Redaksjonen. 'Diabetes type 1'. (2021), [Online]. Available: <https://www.helsebiblioteket.no/pasientinformasjon/alle-brosjyrer/diabetes-type-1> (visited on 29th Mar. 2022).
- [7] M. A. Atkinson, G. S. Eisenbarth and A. W. Michels, 'Type 1 diabetes', *The Lancet*, vol. 383, no. 9911, pp. 69–82, 2014, ISSN: 0140-6736. DOI: [https://doi.org/10.1016/S0140-6736\(13\)60591-7](https://doi.org/10.1016/S0140-6736(13)60591-7). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0140673613605917>.
- [8] Helsedirektoratet. 'Diabetes type 1'. (2020), [Online]. Available: <https://www.helsenorge.no/sykdom/diabetes/diabetes-type-1/> (visited on 1st Apr. 2022).
- [9] B. O. Åsvold. 'Diabetes type 1'. (2021), [Online]. Available: [https://sml.sn.no/diabetes\\_type\\_1](https://sml.sn.no/diabetes_type_1) (visited on 26th Apr. 2022).
- [10] D. Daneman, 'Type 1 diabetes', *The Lancet*, vol. 367, no. 9513, pp. 847–858, 2006, ISSN: 0140-6736. DOI: [https://doi.org/10.1016/S0140-6736\(06\)68341-4](https://doi.org/10.1016/S0140-6736(06)68341-4). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0140673606683414>.
- [11] B. O. Åsvold. 'Blodsukker'. (2021), [Online]. Available: <https://sml.sn.no/blodsukker> (visited on 28th Apr. 2022).
- [12] M. Kvam. 'Insulinsjokk: Dette skjer med hjernen'. (2021), [Online]. Available: <https://nhi.no/sykdommer/hormoner-og-naring/diabetes-type-1/insulinsjokk-og-hjernen/> (visited on 28th Apr. 2022).
- [13] B. O. Åsvold, J. Halse and J. P. Berg. 'Hba1c'. (2021), [Online]. Available: <https://sml.sn.no/HbA1c> (visited on 19th Apr. 2022).
- [14] D. B. Little Randie Ra; Sacks, 'Hba1c: How do we measure it and what does it mean?', *Current Opinion in Endocrinology, Diabetes and Obesity*, vol. 16, no. 2, pp. 113–118, 2009. DOI: 10.1097/MED.0b013e328327728d.
- [15] K. F. Løvaas, T. V. Madsen, U. G. Åstrøm, S. Sandberg and J. Cooper, 'Hba1c, langtidsblodsukker', 2020. [Online]. Available: <https://www.noklus.no/media/xaoh11f2/a-rsrapport-norsk-diabetesregister-for-voksne-2019.pdf> (visited on 11th May 2022).
- [16] N. Helseinformatikk. 'Hba1c, langtidsblodsukker'. (2021), [Online]. Available: <https://nhi.no/sykdommer/hormoner-og-naring/diabetes-generelt/hba1c/> (visited on 11th May 2022).

- 
- [17] J. Halse. 'Diabetiske senkomplikasjoner'. (2019), [Online]. Available: [https://sml.snl.no/diabetiske\\_senkomplikasjoner](https://sml.snl.no/diabetiske_senkomplikasjoner) (visited on 19th Apr. 2022).
- [18] Diabetesforbundet. 'Blodsukker og måling'. (2022), [Online]. Available: <https://www.diabetes.no/diabetes-type-1/behandling/blodsukker/> (visited on 29th Apr. 2022).
- [19] T. B. Vikøren, 'Feilkilder ved bruk av hba1c', [Online]. Available: <https://www.bioingenioren.no/contentassets/4fe8afb09d404189bc10a010c4bc1662/feilkilder-ved-bruk-av-hba1c.pdf> (visited on 28th Apr. 2022).
- [20] W. G. Blackard, C. O. Barlascini, J. N. Clore and J. E. Nestler, 'Morning Insulin Requirements: Critique of Dawn and Meal Phenomena', *Diabetes*, vol. 38, no. 3, pp. 273–277, Mar. 1989, ISSN: 0012-1797. DOI: 10.2337/diab.38.3.273. eprint: <https://diabetesjournals.org/diabetes/article-pdf/38/3/273/356816/38-3-273.pdf>. [Online]. Available: <https://doi.org/10.2337/diab.38.3.273>.
- [21] M. F. Carroll and D. S. Schade, 'The dawn phenomenon revisited: Implications for diabetes therapy', *Endocrine Practice*, vol. 11, no. 1, pp. 55–64, 2005, ISSN: 1530-891X. DOI: <https://doi.org/10.4158/EP.11.1.55>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1530891X20399572>.
- [22] Diabetesforbundet. 'Høyt blodsukker (hyperglykemi)'. (2022), [Online]. Available: <https://www.diabetes.no/diabetes-type-1/behandling/blodsukker/hoyt-blodsukker-hyperglykemi/> (visited on 21st Apr. 2022).
- [23] D. A. Rawshani. 'Kolhydraträkning: 500-regeln, 300-regeln och 100-regeln för att dosera insulin'. (2017), [Online]. Available: <https://diabetes.nu/kolhydratrakning-500-300-100-regeln-dosera-insulin/> (visited on 22nd Apr. 2022).
- [24] E. S. Nilsen, U. Uhrskov, I. W. Rønning and P. Holmquist. 'Diabetes type 1 hos barn'. (), [Online]. Available: [https://finnmarkssykehuset.no/Documents/Pasientbrosjyrer/Diabetes%5C%20type%5C%201%5C%20hos%5C%20barn\\_12s.pdf](https://finnmarkssykehuset.no/Documents/Pasientbrosjyrer/Diabetes%5C%20type%5C%201%5C%20hos%5C%20barn_12s.pdf) (visited on 21st Apr. 2022).
- [25] Helsedirektoratet. 'Diabetes og blodsukker ved sykdom og stress'. (2018), [Online]. Available: <https://www.helsenorge.no/sykdom/diabetes/diabetes-og-andre-sykdommer/> (visited on 28th Apr. 2022).
- [26] —, 'Diabetes, trening og kosthold'. (2018), [Online]. Available: <https://www.helsenorge.no/sykdom/diabetes/diabetes-trening-og-kosthold/> (visited on 28th Apr. 2022).
- [27] A. Advani, 'Positioning time in range in diabetes management', *Diabetologia* 63, pp. 242–252, 2020. DOI: <https://doi.org/10.1007/s00125-019-05027-0>.
- [28] Diabetesforbundet. 'Insulin'. (2021), [Online]. Available: <https://www.diabetes.no/diabetes-type-1/behandling/insulin/> (visited on 26th Apr. 2022).
- [29] E. M. Agency, 'Novorapid : Epar - product information', 2016. [Online]. Available: [https://ec.europa.eu/health/documents/community-register/2004/200404077828/anx\\_7828\\_en.pdf](https://ec.europa.eu/health/documents/community-register/2004/200404077828/anx_7828_en.pdf) (visited on 26th Apr. 2022).
- [30] B. O. Åsvold. 'Insulin'. (2021), [Online]. Available: <https://sml.snl.no/insulin> (visited on 26th Apr. 2022).
- [31] J. L. Bernal, S. Cummins and A. Gasparrini, 'Interrupted time series regression for the evaluation of public health interventions: a tutorial', *International Journal of Epidemiology*, vol. 46, no. 1, pp. 348–355, Jun. 2016, ISSN: 0300-5771. DOI: 10.1093/ije/dyw098. eprint: <https://academic.oup.com/ije/article-pdf/46/1/348/24170929/dyw098.pdf>. [Online]. Available: <https://doi.org/10.1093/ije/dyw098>.
-

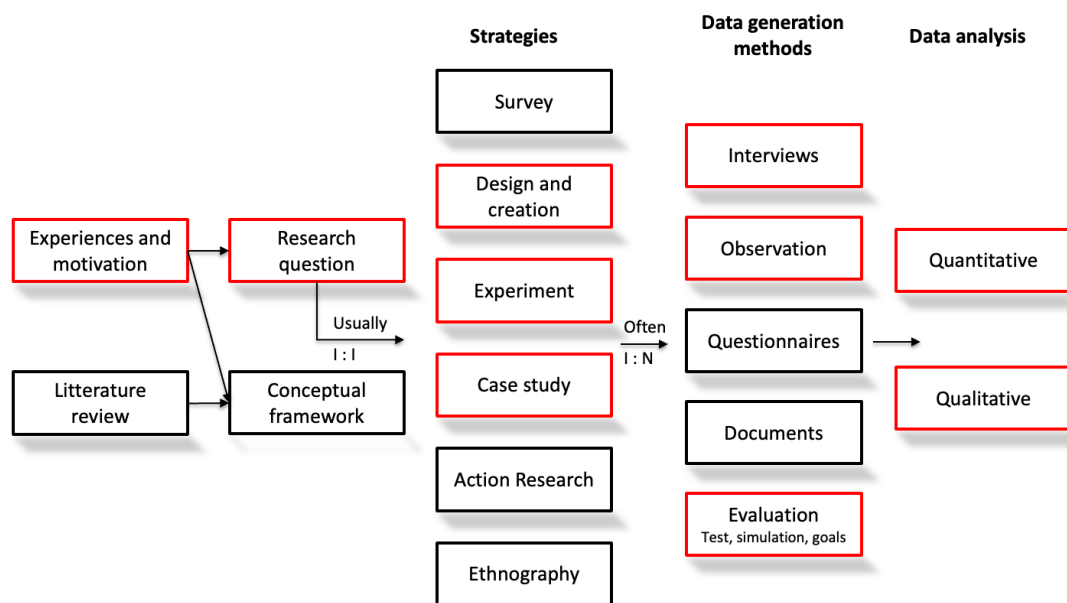
- 
- [32] T. Sauer, *Numerical Analysis*, 2nd ed. Pearson, 2014.
- [33] E. Kant, 'Understanding and automating algorithm design', *IEEE Transactions on Software Engineering*, vol. SE-11, no. 11, pp. 1361–1374, 1985. DOI: 10.1109/TSE.1985.231884.
- [34] N. Tesdal, 'Systemutviklingsmetodikk', *Institutt for datateknologi og informatikk, NTNU, IDATT1002 Systemutvikling*, 2019.
- [35] K. Beck, M. Beedle, A. Bennekum et al. 'Manifesto for agile software development'. (2001), [Online]. Available: <https://agilemanifesto.org/iso/en/manifesto.html> (visited on 1st Apr. 2022).
- [36] A. Wrålsen, 'Interaksjonsdesign som prosess 2', *TDAT2003 Systemutvikling 2 med web-applikasjoner*, 2016.
- [37] R. B. Penfold and F. Zhang, 'Use of interrupted time series analysis in evaluating health care quality improvements', *Academic Pediatrics*, vol. 13, no. 6, Supplement, S38–S44, 2013, Quality Improvement in Pediatric Health Care, ISSN: 1876-2859. DOI: <https://doi.org/10.1016/j.acap.2013.08.002>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1876285913002106>.
- [38] K. Donsa, P. Beck, J. Plank et al., 'A toolbox to improve algorithms for insulin-dosing decision support', *National Library of Medicine*, vol. 5, no. 2, pp. 548–556, 2014. DOI: <https://doi.org/10.4338/ACI-2014-04-RA-0033>. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4081755/>.
- [39] A. E. Fathi, R. E. Kearney, E. Palisaitis, B. Boulet and A. Haidar, 'A model-based insulin dose optimization algorithm for people with type 1 diabetes on multiple daily injections therapy.', *National Library of Medicine*, vol. 68, no. 4, pp. 1208–1219, 2021. DOI: <https://doi.org/10.1109/TBME.2020.3023555>. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4081755/>.
- [40] S. Grønmo. 'Forskningsmetode - samfunnsvitenskap'. (2021), [Online]. Available: [https://snl.no/forskningsmetode\\_-\\_samfunnsvitenskap](https://snl.no/forskningsmetode_-_samfunnsvitenskap) (visited on 5th Apr. 2022).
- [41] A. K. Wagner, S. B. Soumerai, F. Zhang and D. Ross-Degnan, 'Segmented regression analysis of interrupted time series studies in medication use research', *Journal of Clinical Pharmacy and Therapeutics*, 2002. DOI: <https://doi.org/10.1046/j.1365-2710.2002.00430.x>.
- [42] B. Eisenman, *Learning React Native: Building Native Mobile Apps with JavaScript*, 1st ed. O'Reilly Media, 2015.
- [43] Microsoft. 'Using f with xamarin'. (2021), [Online]. Available: <https://docs.microsoft.com/en-us/xamarin/cross-platform/platform/fsharp/> (visited on 11th May 2022).
- [44] S. Bhosale, T. Patil and P. Patil, 'Sqlite: Light database system', *International Journal of Computer Science and Mobile Computing*, vol. 4, no. 4, pp. 882–885, 2015.
- [45] B. Svihus and B. Lande. 'Karbohydrater (ernæring)'. (2021), [Online]. Available: [https://sml.snl.no/karbohydrater\\_-\\_ern%C3%A6ring](https://sml.snl.no/karbohydrater_-_ern%C3%A6ring) (visited on 9th May 2022).
- [46] A. Nes, 'Etikk og ingeniørvirksomhet 3: Ingeniørers profesjonsetikk', 2019.
- [47] NITO. 'Hvorfor er etikk viktig for medlemmer, tillitsvalgte og ansatte i nito?' (2019), [Online]. Available: <https://www.nito.no/organisasjon/om-nito/etikk/> (visited on 13th Apr. 2022).
- [48] Tekna. 'Etikk'. (2021), [Online]. Available: <https://www.tekna.no/om-tekna/etikk/> (visited on 13th Apr. 2022).
-

- 
- [49] K. Malterud, *Kvalitative metoder i medisinsk forskning*, 2nd ed. Universitetsforlaget, 2003.
- [50] Wikipedia. 'Bærekraft'. (2022), [Online]. Available: <https://no.wikipedia.org/wiki/B%C3%A6rekraft> (visited on 12th Apr. 2022).
- [51] A. Tjernshaugen. 'Bærekraft'. (2022), [Online]. Available: <https://snl.no/b%C3%A6rekraft> (visited on 12th Apr. 2022).
- [52] V. G. Moshnyaga, 'Lifecycle energy assessment of mobile applications', in *Proceedings of the 2013 International Workshop on Software Development Lifecycle for Mobile*, ser. DeMobile 2013, Saint Petersburg, Russia: Association for Computing Machinery, 2013, pp. 17–23, ISBN: 9781450323123. DOI: 10.1145/2501553.2501557. [Online]. Available: <https://doi.org/10.1145/2501553.2501557>.
- [53] F.-s. 2020. 'Co2-utslipp per innbygger i norge'. (2018), [Online]. Available: <https://www.fn.no/Statistikk/co2-utslipp-per-innb?country=306> (visited on 18th May 2022).
- [54] FN-sambandet. 'God helse og livskvalitet'. (2022), [Online]. Available: <https://www.fn.no/om-fn/fns-baerekraftsmaal> (visited on 8th Apr. 2022).
- [55] HELFO. 'Frikort for helsetjenester'. (2022), [Online]. Available: <https://www.helsenorge.no/betaling-for-helsetjenester/frikort-for-helsetjenester/> (visited on 18th May 2022).
- [56] B. J. Oates, M. Griffiths and R. McLean, *Researching Information Systems and Computing*, 2nd ed. SAGE, 2022.
- [57] R. Goodwin. 'Iphone battery sizes (mah) – iphone 3gs to iphone 13'. (2021), [Online]. Available: <https://www.knowyourmobile.com/user-guides/iphone-battery-sizes/> (visited on 19th Apr. 2022).
- [58] Folkehelseinstituttet. 'Nye tall om hvor mange som har diabetes i norge'. (2020), [Online]. Available: <https://www.fhi.no/nyheter/2020/nye-tall-om-hvor-mange-som-har-diabetes-i-norge/> (visited on 19th Apr. 2022).
- [59] N. vassdrags- og energidirektorat. 'Lavt klimagassutslipp knyttet til norsk strømforbruk i 2020'. (2020), [Online]. Available: <https://www.nve.no/nytt-fra-nve/nyheter-energi/lavt-klimagassutslipp-knyttet-til-norsk-stromforbruk-i-2020/> (visited on 18th May 2022).



# Vedlegg

## A Forskningsmetode

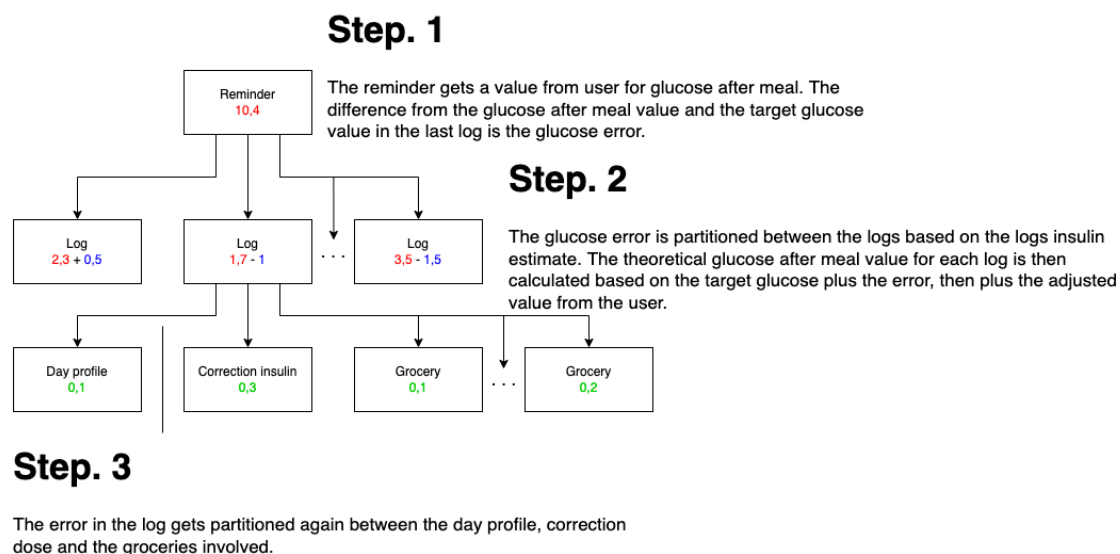


(basert på Fig.3.1 Model of the research process, Oates 2006, p33)

Figur 17: Forskningsmetode

Figur 17 er hentet fra boken «Researching Information Systems and Computing» [56]. De røde boksene er de gruppen har valgt å benytte seg av.

## B Algoritmen visualisert



Figur 18: Algoritmen visualisert

## C Livsløpsanalyse og beregninger

MVP-en bruker en lokale database, det betyr at utslippene fra denne løsningen kun kommer av produksjonen av mobile enheter, samt strømforbruket til MVP-en.

Det er trygt å anta at dersom MVP-en som ferdig applikasjon blir tilgjengelig på alle mobile plattformer så vil det ikke være noen tilfeller av at noen kjøper en mobil enhet for å bruke applikasjonen. Dette er noe de vil ha fra før. Derfor ses det bort ifra livsløpet til den mobile enheten, da dette kan variere mye også.

Det er da eksklusivt strømforbruket som er en faktor. Dersom man observerer batteriforbruket til MVP-en, vil man se at den, med data fra de 10 siste dagene, bruker omtrent 2% av mobilens totale strømforbruk. Denne dataen er hentet direkte fra den mobile enheten til testpersonen. Testpersonen sin mobile enhet, en iPhone 11, har en batterikapasitet på 3110 mAh [57]. Mobilen har også en batteri helse på 87% av et nytt batteri, dette er hentet fra «Battery Health» på testpersonens mobil. Dette gir en total kapasitet på  $3110mAh * 0,87 \approx 2706mAh$ . For å lade telefonen til 100%, tar det omtrent  $P = U * I = 5V * 2706mAh = 13,53Wh$ . 2% av dette er omtrent  $13,53Wh * 0,02 = 0,27Wh$ . Dersom man antar at forbruket forholder seg likt i løpet av ett år, vil den totale mengden være omtrent  $0,22Wh * 365dager = 80Wh/r$ .

I 2020 faststo FHI at det er omtrent 316.000 - 345.000 nordmenn med diabetes. Av disse er det omtrent 10 prosent som har diabetes type 1 [58]. Dette tilsvarer omtrent  $\frac{316.000+345.000}{2} (gjennomsnitt) * 0,1 = 33.050$  personer med diabetes type 1 i Norge. I det ekstreme tilfellet der alle nordmenn med diabetes type 1 hadde tatt i bruk MVP-en, ville dette medført et strømforbruk på  $80Wh/aar * 33.050personer = 2.644.000Wh/aar = 2.644kWh/aar$ .

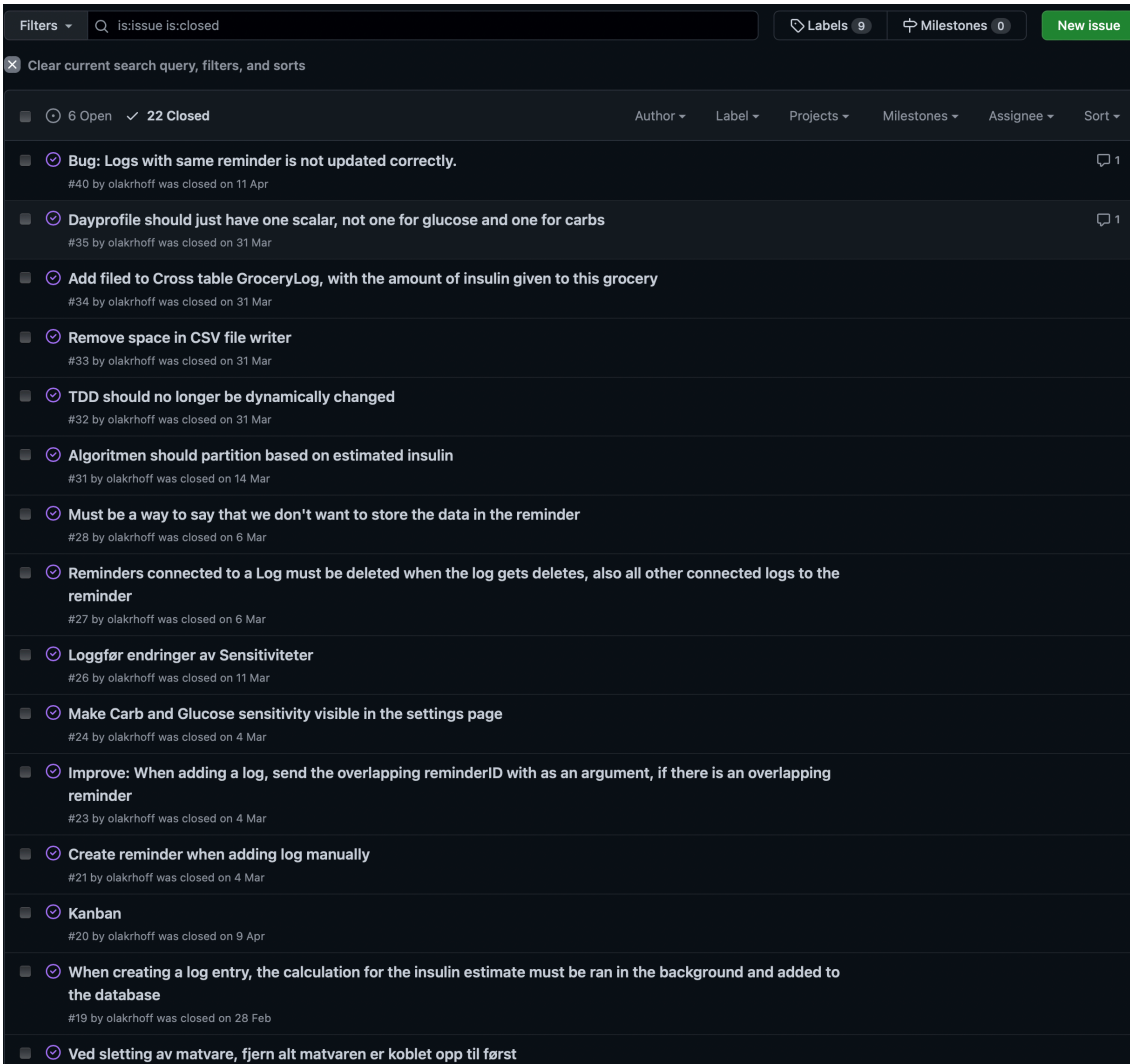
Hvor mye  $CO_2$  dette tilsvarer i utslipp varierer på hvor strømmen kommer fra. I Norge, for å fortsette eksempelet, var det i 2020 omtrent 8 gram  $CO_2$ -ekvivalenter/kWh [59].

Dette vil si at vi får utslipp på omtrent  $2.644kWh/aar * 8gram CO_2 - ekvivalenter/kWh = 21,152kg CO_2 - ekvivalenter/aar \approx 21kg CO_2 - ekvivalenter/aar$ .

## D Vedlagte dokumenter i prosjektmappe

- Forprosjektplan
- Visjonsdokument
- Systemdokumentasjon
- Kravdokumentasjon
- Prosessdokumentasjon (prosjekthåndbok)
- Presentasjon av bacheloroppgave

## E Issue-board



The screenshot shows a Jira issue board with the following details:

- Filters:** is:issue is:closed
- Labels:** 9
- Milestones:** 0
- Buttons:** Clear current search query, filters, and sorts; New issue
- Summary:** 6 Open, 22 Closed
- Columns:** Author, Label, Projects, Milestones, Assignee, Sort
- Issues (from top to bottom):**
  - #40 by olakrhoff was closed on 11 Apr: Bug: Logs with same reminder is not updated correctly. (1 comment)
  - #35 by olakrhoff was closed on 31 Mar: Dayprofile should just have one scalar, not one for glucose and one for carbs. (1 comment)
  - #34 by olakrhoff was closed on 31 Mar: Add filed to Cross table GroceryLog, with the amount of insulin given to this grocery
  - #33 by olakrhoff was closed on 31 Mar: Remove space in CSV file writer
  - #32 by olakrhoff was closed on 31 Mar: TDD should no longer be dynamically changed
  - #31 by olakrhoff was closed on 14 Mar: Algoritmen should partition based on estimated insulin
  - #28 by olakrhoff was closed on 6 Mar: Must be a way to say that we don't want to store the data in the reminder
  - #27 by olakrhoff was closed on 6 Mar: Reminders connected to a Log must be deleted when the log gets deletes, also all other connected logs to the reminder
  - #26 by olakrhoff was closed on 11 Mar: Loggfør endringer av Sensitiviteter
  - #24 by olakrhoff was closed on 4 Mar: Make Carb and Glucose sensitivity visible in the settings page
  - #23 by olakrhoff was closed on 4 Mar: Improve: When adding a log, send the overlapping reminderID with as an argument, if there is an overlapping reminder
  - #21 by olakrhoff was closed on 4 Mar: Create reminder when adding log manually
  - #20 by olakrhoff was closed on 9 Apr: Kanban
  - #19 by olakrhoff was closed on 28 Feb: When creating a log entry, the calculation for the insulin estimate must be ran in the background and added to the database
  - Ved sletting av matvare, fjern alt matvaren er koblet opp til først

Figur 19: Issue-board

---

## F Brukertester

### Fremgangsmåten for brukertestene:

Brukertesten skal være semi-strukturert. Bruker skal utforske siden på egenhånd, men dersom bruker ikke finner all funksjonalitet, skal testansvarlig stille oppfølgingsspørsmål. Før man lar bruker starte med å teste funksjonalitet skal **KALKULATOR SIDE** spørsmålene stilles. **KALKULATOR SIDE** og **SPØRSMÅL ETTER TESTEN** må stilles, mens **SPØRSMÅL ANGÅENDE FUNKSJONALITET SOM BRUKER IKKE FANT PÅ EGENHÅND** skal kun stilles dersom bruker ikke har funnet all funksjonalitet. Husk å notere svar.

### F.1 Brukertest 1 - Ikke diabetiker

Bruker: Kvinne, 24 år, har ikke diabetes.

Denne brukertesten ble gjennomført for å undersøke om designet til MVP'en er intuitivt, selv uten erfaringer med diabetes.

KALKULATOR SIDE (HOVEDSIDE):

#### Er siden oversiktlig?

- Ser ut som DNB appen
- Behagelig farger
- Ser profesjonell ut
- Litt tom
- Fin meny
- Oversiktlig
- Fine ikoner

#### Hva tror du du kan gjøre med applikasjonen/MVP'en?

- Ser ut som en økonomi app
- Minner om bank

#### Hva tror du tabben "logg" består av?

- Det man har logget tidligere, og det man kan logge.

#### Hva tror du tabben "matvarer" består av?

- Man kan velge matvarer eller logge matvarer
- Trykket inn og så at man kunne registrere matvarer, synes dette ga mening.

#### Hva tror du tabben "dagsprofiler" består av?

- Målingene dine for dagen

- 
- Trykket inn og så at man kunne opprette dagsprofiler. Grunnet at brukeren ikke har diabetes og ikke kan mye om sykdommen, var hun ikke klar over at man kan ha ulik sensitivitet for insulin basert på tidspunkt på dagen. Testansvarlig forklarte fenomenet til bruker, slik at funksjonaliteten skulle gi mer mening etter at bruker hadde gitt sine tanker, og brukertesten var over.

#### SPØRSMÅL ANGÅENDE FUNKSJONALITET SOM BRUKER IKKE FANT PÅ EGENHÅND (EKSEMPLER PÅ OPPFØLGINGSSPØRSMÅL):

Bruker fikk ikke testet mye funksjonalitet på egenhånd, da matvarer og dagsprofiler måtte på plass før man kunne kalkulere insulin. Bruker undersøkte de ulike tabbene i menyen, og trykket på "calculate" på kalkulator siden, og fikk opp feilmelding om at man må ha den nåværende blodsukker verdien for å kalkulere. Trykket på "add matvarer"-knappen, forventet å legge til matvarer. Prøvde å søke etter matvarer, innså at det ikke eksisterte noen.

#### **Hva tror du det øverste feltet på kalkulator siden er?**

- Til å skrive inn current glucose/blodsukker verdien.
- Trykket inn og så at man kunne registrere matvarer, synes dette ga mening.

#### **Hvordan ville du regnet ut hvor mange insulinenheter du må sette til et måltid?**

- La til current glucose og matvarer, brunch ble valgt automatisk.
- Trykte inn i feltet ved units, antok man kunne override tallet hvis man var uenig.

#### **Hva ville du gjort for å opprette en ny matvare?**

- Trykket på pluss tegnet. La til brød og fylte ut feltene. Trykket på save-knappen

#### **Hvordan ville du endret hvilke matvarer du skulle spise til et gitt måltid?**

- Trykket på "add groceries"-knappen, la til brød.

#### **Hvordan ville du undersøkt om du allerede hadde en eksisterende matvare som du ville spise?**

- Trykket groceries tabben, fant brød.

#### **Hvordan ville du slettet en matvare?**

- Swipet på matvaren for å slette den, slik som man gjør på telefonen.

#### **Hvordan ville du opprettet dagsprofiler?**

- Trykket på plusstegnet på dayprofiles. Lagde "brunch"

---

**Hvordan ville du endret dagsprofilen til en måltid, dersom du ville ha en annen en?**

- La til en ny dagsprofil, etter instruksjer fra testansvarlig (dette ble gjort før spørsmålet ble stilt). Trykket på der det sto "second b", navnet på den nye dagsprofilen, på kalkulator siden, og endret til brunch ved bruk av picker funksjonaliteten.

**Hvordan ville du slettet en dagsprofil?**

- Gjorde som med matvare, swipet på dagsprofilen.

**Hva ville du gjort dersom det dukket opp en påminnelse?**

- Trykket "ok". Skrev inn glukose

**Hvis du skal spise en matvare f.eks. 2 timer etter sist gang du registrerte insulin i applikasjonen, hva ville du gjort?**

- Bruker observerte at man ikke kunne skrive inn "current glucose". Trykket på "add groceries"-knappen. Trykket på calculate-knappen.

**Hvordan vil du sjekke hva du spiste til et måltid?**

- Gikk på logg-tabben, trykket på en logg og fant informasjon om denne.

**SPØRSMÅL ETTER TESTEN:**

**Var applikasjonen lett/komfortabel å bruke?**

- Ja.
- Bruker mente det ville vært lettere hvis hun visste mer om diabetes som helhet.
- Følt intuitiv.
- Typisk app, når hun trykker på noe forventer hun at det er det som skal skje.
- Oversiktlig, og applikasjonen var ikke crowded/rotete. Nøytral applikasjon. Fint design. Ingen surrete fonter. Fine ikoner.

**Var det enkelt å forstå hvordan og hva du skulle gjøre?**

- Ja, stort sett.
- Det som holdt bruker tilbake var en bedre forståelse av diabetes og kunnskaper innen feltet

**Var informasjonen på sidene enkel å forstå, og var funksjonaliteten til applikasjonen formidlet på en god måte?**

- Ja, relativt åpenbart

---

**Hva syns du om designet? Burde noe vært et annet sted? Farger?**

- Fargene er nøytrale og behagelige. Bruker mente det ville vært rart å ha andre farger.
- Applikasjonen minner om bank app, men bruker synes det ga mening, da man utfører en slags økonomi her og.
- Appen er simpel og brukervennlig.
- Bruker syns det mangler en kul logo.

**Når du trykket på de forskjellige knappene/sidene/utforsket funksjonaliteten, var det som skjedde, det du forventet skulle skje? Skjedde noe uventet? Var noe frustrerende, eller var noen av navnene på knappene/funksjonaliteten misvisende?**

- Ja, funksjonaliteten var forutsigbar.
- Bruker synes hun visste lite, men syns alt gir mening av knapper og den type funksjonalitet.

**Hvordan føler du deg etter å ha brukt applikasjonen? Nøytral, frustrert, fornøyd, produktiv?**

- Litt forvirret, men bruker mente det var hennes manglende kunnskaper innen diabetes.
- Følte seg effektiv.

**Hva er din helhetlig mening angående applikasjonen? Ville du anbefalt den til andre?**

- Ville anbefalt den til andre.
- Bruker syns det virker som et fornuftig verktøy.

**Syns du ikonene matchet med teksten som sto ved siden av? Var de misvisende? Burde det vært et annet ikon eller tekst?**

- Ja, syntes ikoner og tekst gikk overens.

---

## F.2 Brukertest 2 - Diabetiker

Bruker: Kvinne, 20 år, har diabetes.

KALKULATOR SIDE (HOVEDSIDE):

### Er siden oversiktlig?

- Ja.

### Hva tror du du kan gjøre med applikasjonen/MVP'en?

- Kan sette inn verdier for blodsukker. Regne ut insulin for karbohydrater.

### Hva tror du tabben "logg" består av?

- Oversikt over tidligere spiste matvarer og antall, blodsukker, osv.

### Hva tror du tabben "matvarer" består av?

- Matvarer.

### Hva tror du tabben "dagsprofiler" består av?

- Det som har blitt spist i løpet av en dag, hvordan den har estimert enhetene hennes.

SPØRSMÅL ANGÅENDE FUNKSJONALITET SOM BRUKER IKKE FANT PÅ EGENHÅND (EKSEMPLER PÅ OPPFØLGINGSSPØRSMÅL):

### Hvordan ville du regnet ut hvor mange insulinenheter du må sette til et måltid?

- La inn hvilket blodsukker hun hadde akkurat. La til matvarer. Trykket på Calculate. Trykket på «Log insulin»-knapp.

### Hva ville du gjort for å opprette en ny matvare?

- Var inne på Add groceries. Trykket på «+»-tegnet øverst til høyre. La inn en matvare Pitabrød.

### Hvordan ville du endret hvilke matvarer du skulle spise til et gitt måltid?

- Trykket på Add groceries, og endret matvarer.

### Hvordan ville du undersøkt om du allerede hadde en eksisterende matvare som du ville spise?

- Gikk på Groceries tabben. Lurte på om det var søkefunksjonalitet på siden, det er det ikke.



---

### **Hvordan ville du slettet en matvare?**

- Swipet til høyre og trykket på Delete på matvaren.

### **Hvordan ville du opprettet dagsprofiler?**

- Gikk på Day profiles. Trykket på «+»-tegnet. Skrev inn i feltene og la til Brunch.

### **Hvordan ville du endret dagsprofilen til en måltid, dersom du ville ha en annen en?**

- Trykket der det sto «Middag» (dukket opp automatisk grunnet tidspunkt og at den var lagt til som en Day profile). Endret til «Brunch»

### **Hvordan ville du slettet en dagsprofil?**

- Swipet til høyre og trykket på Delete på Day profile'en

### **Hva ville du gjort dersom det dukket opp en påminnelse?**

- Trykket «OK». La inn hva hun ble etter tre timer. Trykket «OK».

### **Hvis du skal spise en matvare f.eks. 2 timer etter sist gang du registrerte insulin i applikasjonen, hva ville du gjort?**

- La til flere matvarer, fikk ikke endret blodsukker verdi fra ønsket verdi fra sist (overlappende måltid).

### **Hvordan vil du sjekke hva du spiste til et måltid?**

- Gikk til Log tabben. Trykket på logg-entriet.

### **SPØRSMÅL ETTER TESTEN:**

#### **Var applikasjonen lett/komfortabel å bruke?**

- Ja.

#### **Var det enkelt å forstå hvordan og hva du skulle gjøre?**

- Ja, sto tydelig hva man kunne gjøre.

#### **Var informasjonen på sidene enkel å forstå, og var funksjonaliteten til applikasjonen formidlet på en god måte?**

- Ja, synes det. Var lett å forstå hva man skulle og kunne gjøre.

#### **Hva syns du om designet? Burde noe vært et annet sted? Farger?**

- 
- Designet var enkelt og pent, lett å se på. Fine farger.

**Når du trykket på de forskjellige knappene/sidene/utforsket funksjonaliteten, var det som skjedde, det du forventet skulle skje? Skjedde noe uventet? Var noe frustrerende, eller var noen av navnene på knappene/funksjonaliteten misvisende?**

- Ja, det var intuitivt hva som skulle skje. Ingenting uventet.

**Hvordan føler du deg etter å ha brukt applikasjonen? Nøytral, frustrert, fornøyd, produktiv?**

- Spent, gleder seg til andre, og seg selv, kan få lov til å bruke applikasjonen. Fornøyd med hvordan applikasjonen virker så langt.

**Hva er din helhetlig mening angående applikasjonen? Ville du anbefalt den til andre?**

- Ja, ville anbefalt den. Spesielt til de som har penn/sprøyte for å sette insulin. Hennes pumpe har deler av funksjonaliteten i applikasjonen, og derfor tenker hun det er spesielt fint for de som ikke har dette, som for eksempel de som bruker penn. Det finnes flere applikasjoner for diabetes, men denne har samlet alt det nødvendige. Det er greit å ha alt på en app.

**Syns du ikonene matchet med teksten som sto ved siden av? Var de misvisende? Burde det vært et annet ikon eller tekst?**

- Synes det var enkelt og greit, og det matchet.

