

UNIVERSITEIT TWENTE.

Een stoplichtsysteem voor de monitoring van bloedglucoseregulatie bij diabetes type 2 patiënten

Bacheloropdracht Biomedische Technologie

Elling, R.A. (Student B-BMT)

03-07-2024

Afstudeercommissie:

Prof.dr. G.D. Laverman

Dr. A. Middelweerd

Dr. N. Beerlage - de Jong

S.E.M. Teunissen, MSc

Abstract

Nederlands

Om grip te krijgen op de bloedglucoseregulatie bij diabetes type 2 patiënten wordt er gebruik gemaakt van continue glucosemonitoring (CGM). Een CGM-apparaat meet voortdurend de glucoseconcentratie en slaat de waarden op, waarna er na verloop van tijd een Ambulatory Glucose Profile (AGP) kan worden gegenereerd. Met behulp van de parameters van dit AGP-rapport, de High en Low Blood Glucose Index (HBGI/LBGI) en de kleuren van het stoplicht kan er een systeem ontworpen worden dat de actuele glucoseregulatie voor elke patiënt duidelijk weergeeft. Dit 'stoplichtsysteem' kan de zorgprofessional helpen inzicht te krijgen in het gezondheidsrisico van de patiënten, waarna de behandeling hierop aangepast kan worden. Een conceptversie van dit systeem is ontwikkeld, maar de toevoeging van andere relevante data, integratie met huidige technologieën en toepassing binnen de zorg moet nog worden onderzocht.

English

To gain control over blood glucose regulation in patients with type 2 diabetes, continuous glucose monitoring (CGM) is used. A CGM device continuously measures glucose concentration and stores the values, allowing an Ambulatory Glucose Profile (AGP) to be generated over time. Using the parameters of this AGP report, the High and Low Blood Glucose Index (HBGI/LBGI), and traffic light colors, a system can be designed to clearly display the current glucose regulation for each patient. This 'traffic light system' can help healthcare professionals gain insight into the health risks of patients, allowing treatment to be adjusted accordingly. A concept version of this system has been developed, but the addition of other relevant data, integration with current technologies, and application within healthcare still need to be explored.

Inhoudsopgave

Abstract	1
1 Inleiding	3
2 Achtergrondliteratuur	5
2.1 Streefwaarden continue glucosemonitoringdata	5
2.2 Glycemia Risk Index	6
2.3 Low/High Blood Glucose Index	6
2.4 Q-Score	7
3 Methode	8
3.1 Parameterkeuze en kleurkoppeling	8
3.2 Bepaling van de tijdsperiode	9
3.3 Waarschuwingen	10
3.4 Dataverwerking en weergave	10
4 Systeemontwerp	11
5 Systeemconcept	13
6 Scenario's en persona's	15
6.1 Scenario 1: Johan de Vries - verpleegkundige	15
6.2 Scenario 2: Dirk Jansen - endocrinoloog	15
6.3 Scenario 3: André de Jong - patiënt	17
7 Conclusies en aanbevelingen	19
A app.R	25
B functions.R	40

1 Inleiding

Diabetes mellitus type 2 patiënten lopen een verhoogd risico op gezondheidsproblemen zoals hart- en vaatziekten. Verschillende factoren zoals gewicht, beweging en voeding dragen bij aan de regulatie van de bloedglucosespiegel en de ernst van de diabetes [1]. Om inzicht te krijgen in de leefstijl van de patiënt is er door Ziekenhuisgroep Twente (ZGT) en Universiteit Twente (UT) de Diameter ontwikkeld. Dit is een programma waarbinnen data wordt verzameld over bloedglucosewaarden, beweging en voeding met behulp van respectievelijk een continue glucosemonitorsysteem (Abbott FreeStyle Libre), een Fitbit stappenteller en een voedingsdagboek. Met deze gegevens kan er een persoonlijk leefstijladvies gegeven worden. Aan de hand van de kwaliteit van de bloedglucoseregulatie kan het gezondheidsrisico van de patiënt worden beoordeeld. Hiervoor is er vanuit het International Diabetes Center een gestandaardiseerd glucoseprofiel, Ambulatory Glucose Profile (AGP), ontwikkeld. Bij het gebruik van een continue glucosemonitor (CGM) kan dit rapport worden gegenereerd. In het geval van de Freestyle Libre is dit mogelijk in de bijbehorende web-applicatie LibreView. Het rapport kan door de zorgprofessional en de patiënt worden geanalyseerd, waarna er adviezen voor verandering in leefstijl en/of behandeling gegeven kunnen worden. De richtlijn is om hiervoor 14 dagen aan data te verzamelen. Hiermee kan namelijk een goede schatting van de glucoseregulatie over een periode van drie maanden worden gemaakt. Na deze periode is het van belang een nieuw rapport te genereren [2].

Een belangrijke maatstaf in dit rapport is de Time in Range (TIR), wat het percentage van de tijd aangeeft dat de bloedglucoseconcentratie binnen een bepaald bereik valt. Voor de TIR wordt er gestreefd naar minimaal 70% van de tijd een glucosewaarde tussen 3,9 en 10,0 mmol/L, zoals te zien is in figuur 1.1. De TIR correleert namelijk met de HbA1c-waarde, die de glucoseregulatie over een periode van 3 maanden weerspiegelt [3].

Om de mogelijkheid tot integratie van de Diameter in de zorg te onderzoeken, is in 2024 de PerfectFit studie gestart. Het doel hierin is het aanbieden van een combinatie van face-to-face en digitale zorg (blended care). Binnen de diabeteszorg zou hiermee de werkdruk verminderd en de kosten verlaagd kunnen worden [4]. Hierbij is het van belang dat de zorgprofessional de

AGP Report

Name _____

MRN _____

GLUCOSE STATISTICS AND TARGETS

26 Feb 2019–10 Mar 2019 **13 days**
% Time CGM is Active **99.9%**

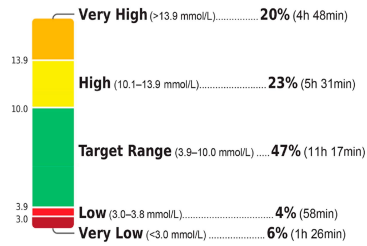
Glucose Ranges	Targets [% of Readings (Time/Day)]
Target Range 3.9–10.0 mmol/L	Greater than 70% (16h 48min)
Below 3.9 mmol/L	Less than 4% (58min)
Below 3.0 mmol/L	Less than 1% (14min)
Above 10.0 mmol/L	Less than 25% (6h)
Above 13.9 mmol/L	Less than 5% (1h 12min)

Each 5% increase in time in range (3.9–10.0 mmol/L) is clinically beneficial.

Average Glucose **9.6 mmol/L**
Glucose Management Indicator (GMI) **7.6%**
Glucose Variability **49.5%**

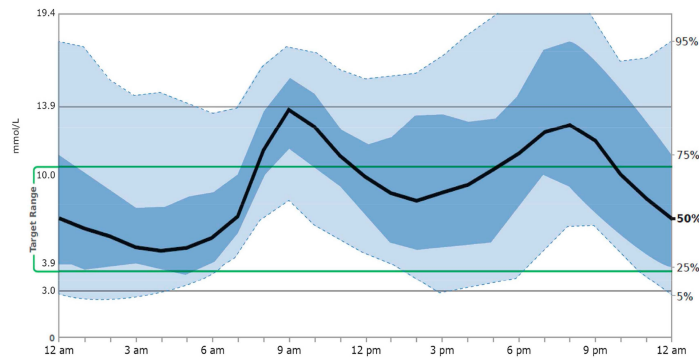
Defined as percent coefficient of variation (%CV); target ≤36%

TIME IN RANGES

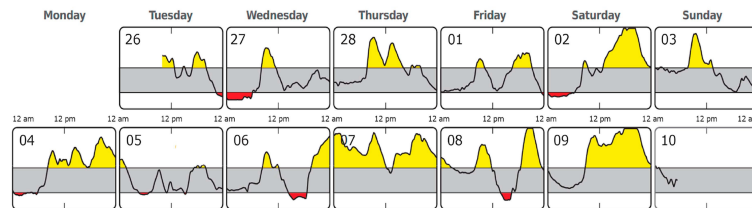


AMBULATORY GLUCOSE PROFILE (AGP)

AGP is a summary of glucose values from the report period, with median (50%) and other percentiles shown as if occurring in a single day.



DAILY GLUCOSE PROFILES



Each daily profile represents a midnight-to-midnight period.

Figuur 1.1: Voorbeeld AGP-rapport. Overgenomen van Battelino et al. [5].

patiënten die meer (of juist minder) zorg behoeven in kaart kan brengen. Een systeem dat het gezondheidsrisico van patiënten op afstand analyseert in combinatie met de weergave hiervan op het dashboard van de zorgprofessional kan hiermee helpen. Onder andere de gegevens van het AGP-rapport en de kleuren van het stoplicht kunnen hiervoor worden gebruikt. Het wordt dan mogelijk om in één oogopslag de huidige glucoseregulatie in te zien, zonder dat men voor elke individuele patiënt een AGP-rapport moet genereren. In dit verslag wordt onderzocht hoe dit 'stoplichtsysteem' vormgegeven en geïmplementeerd kan worden, welke parameters er moeten worden gebruikt en hoe deze te koppelen zijn aan een kleur.

2 Achtergrondliteratuur

2.1 Streefwaarden continue glucosemonitoringdata

De aanbevelingen voor het beoordelen, gebruiken en rapporteren van continue glucose monitoringdata, zijn vastgelegd op basis van internationale consensus [5]. Er is overeengekomen de belangrijkste parameters en streefwaarden vast te leggen in het eerdergenoemde gestandaardiseerde AGP-rapport. Naast de tijd binnen het doelbereik (Time in Range) wordt hierin ook gekeken naar de Time Below Range (<3,9 mmol/L) en Time Above Range (>10,0 mmol/L). Binnen de Time Below Range wordt onderscheid gemaakt tussen Low (3,0-3,8 mmol/L) en Very Low (<3,0 mmol/L). Binnen de Time Above Range wordt onderscheid gemaakt tussen High (10,1-13,9 mmol/L) en Very High (>13,9 mmol/L). Ook worden de gemiddelde glucosewaarde, de Glucose Management Indicator (GMI) en de glucose variabiliteit (GV) vermeld.

De streeftijden zijn onder andere bepaald aan de hand van de directe correlatie tussen TIR en het risico op complicaties zoals diabetische retinopathie [6]. Ook is er een correlatie tussen de TIR en het HbA1c-percentage. Een TIR van 70% komt gemiddeld gezien namelijk overeen met een HbA1c van 53 mmol/L (7%), al kan dit per individu erg afwijken [3]. Een verhoging van de TIR met 5% komt overeen met een verlaging van HbA1c met 0,4% [7]. Voor de TIR geldt daarom dat elke verhoging van 5% klinisch gunstig is. Bij diabetespatiënten wordt een HbA1c van <53 mmol/L beoordeeld als goed, maar een gepersonaliseerde streefwaarde wordt ook belangrijk geacht [8]. Vanaf een HbA1c van 53 mmol/L geldt dat elke stijging van 1% wordt geassocieerd met 38% hoger risico op een macrovasculaire complicatie, 40% hoger risico op een microvasculaire complicatie en een 38% hoger risico op overlijden [9].

De GV wordt weergegeven als percentage van de variatiecoëfficiënt (%CV). Dit is de procentuele standaardafwijking van de gemiddelde glucosewaarde. Uit onderzoek blijkt dat de gemiddelde GV waarde van diabetes type 2 patiënten zonder medicijnen, of met alleen een medicijn zoals metformine, rond 18% ligt [10]. De uiterste waarde gemeten binnen deze groep is 36%. Bij diabetes type 1 patiënten ligt de GV waarde rond 36%. Ook bij diabetes type 2 patiënten die met insuline behandeld worden komt een waarde van >36% vaak voor. De frequentie van

hypoglykemische episodes is aanzienlijk hoger bij de twee laatstgenoemde groepen, wat duidt op een hoger gezondheidsrisico.

2.2 Glycemia Risk Index

Een recente poging om aan de hand van bovenstaande parameters een formule op te stellen waarmee een enkele waarde kan worden berekend voor de kwaliteit van de bloedglucoseregulatie, is gedaan door Klonoff et al. [11]. Binnen het stoplichtsysteem zou deze waarde gekoppeld kunnen worden aan een kleur, zodat in één oogopslag te zien is hoe goed de glucosespiegel gereguleerd is. Er zijn door 330 ervaren klinici 225 AGP rapporten beoordeeld met een cijfer. Vervolgens is hiermee de formule voor de Glycemia Risk Index (GRI) opgesteld. Deze luidt als volgt: $GRI = (3.0 \times VeryLow) + (2.4 \times Low) + (1.6 \times VeryHigh) + (0.8 \times High)$. Hierbij is 0 de beste waarde en 100 de slechtste waarde. Als de GRI boven 100 uitkomt, wordt deze afgekapt naar 100. Voor het AGP rapport in figuur 1.1 zou de GRI 78 zijn.

2.3 Low/High Blood Glucose Index

Een andere methode om de algemene bloedglucoseregulatie te beoordelen is de beter gedocumenteerde Low Blood Glucose Index/High Blood Glucose Index (LBGI/HBGI), beschreven door Kovatchev et al. [12]. De mate waarin de verschillende parameters meewegen in de formule van de GRI komt hiermee overeen. De bloedglucose-schaal is getransformeerd en symmetrisch gemaakt rond de waarde 6,25 mmol/L, waaraan de score 0 is gekoppeld. Aan de waarden 1,1 en 33,3 mmol/L is de score 100 gekoppeld. Dit maakt de formule voor de glucosewaarden op de nieuwe schaal $f(BG) = 1.794 \times ([\ln BG]^{1,026} - 1, 861)$. Met het risico: $r(BG) = 10 \times f(BG)^2$. BG = bloedglucosewaarde in mmol/L.

Stel dat x_1, x_2, \dots, x_n n aantal BG metingen van een patiënt zijn. Dan geldt:

$$LBGI = \frac{1}{n} \sum_{i=1}^n r_l(x_i)$$

waarbij $r_l(BG) = r(BG)$ als $f(BG) \leq 0$. Anders is deze 0.

Voor de HBGI geldt:

$$HBGI = \frac{1}{n} \sum_{i=1}^n r_h(x_i)$$

waarbij $r_h(BG) = r(BG)$ als $f(BG) > 0$. Anders is deze 0.

Voor de LBGI gelden voor de patiënt de volgende risico-categorieën: laag (<2,5), gematigd (2,5-5) en hoog (>5). Voor de HBGI zijn deze laag (<4,5), gematigd (4,5-9) en hoog (>9) [13]. Met

behulp van deze twee parameters, kan de Average Daily Risk Range (ADRR) worden bepaald. De ADRR is het gemiddelde van de som van de maximale waarden van de LBGI en HBGI per dag. Hiervoor gelden opnieuw drie risico-categorieën: laag risico (<20) gematigd risico (20-40) en hoog risico (>40) [14]. Met de ADRR kan dus het gemiddelde van de intensiteit van de pieken worden weergegeven over meerdere dagen. Voor continue glucosemonitoring (CGM) is ook de Hourly Risk Range (HRR) gedefinieerd, deze is hetzelfde als de ADRR maar dan berekend per uur [15].

2.4 Q-Score

Ook de Quality-Score (Q-Score) is een manier om CGM profielen te screenen die behandeling behoeven. Dit is een samengestelde indicator ontwikkeld door Augstein et al. [16]. Deze is ontwikkeld omdat risico-scores, zoals van de HBGI/LBGI, volgens hen in de praktijk vaak onhandig zijn voor gebruik binnen routinematige diabeteszorg. Er is gekeken naar de belangrijkste factoren en de bijbehorende parameters die het glucoseprofiel bepalen. Ook is er onderzocht hoe de parameters met elkaar correleren, waarna er 5 zijn geselecteerd voor de constructie van de Q-Score. De Q-Score is gedefinieerd als:

$$Q\text{-Score} = 8 + \frac{\text{mean} - 7,8}{1,7} + \frac{\text{range} - 7,5}{2,9} + \frac{t_{G<3,9} - 0,6}{1,2} + \frac{t_{G>8,9} - 6,2}{5,7} + \frac{\text{MODD} - 1,8}{0,9}$$

waarbij voor één dag geldt: mean = gemiddelde bloedglucosewaarde, range = verschil tussen min en max glucosewaarde, $t_{G<3,9}$ = tijd onder 3,9 mmol/L (in uren), $t_{G>8,9}$ = tijd boven 8,9 mmol/L (in uren); voor meerdere dagen geldt: MODD = Mean Of Daily Differences (gemiddeld verschil van de metingen, tussen verschillende dagen, op hetzelfde tijdstip).

De Q-score wordt telkens berekend voor drie opeenvolgende dagen, waarbij voor de parameters het gemiddelde wordt genomen over deze drie dagen. Opvallend is de definitie van het hyper-component. Waarden worden namelijk al meegenomen in de berekening als deze boven 8,9 mmol/L zijn, in plaats van 10,0 mmol/L. De Q-Score heeft 5 kwaliteits-categorieën: zeer goed (<4,0), goed (4,0-5,9), voldoende (6,0-8,4), matig (8,5-11,9), slecht (>11,9).

De Q-Score wordt onder andere gebruikt in combinatie met het Karlsburger Diabetes-Management System (KADIS), dat gebruikt wordt om insuline/glucose profielen te simuleren. KADIS is een beslissingsondersteunend systeem dat de zorgprofessional helpt te kiezen welke diabetes type 2 patiënten aanpassingen in de behandeling nodig hebben. Ook worden de mogelijke effecten van de behandelingen met de Q-Score weergegeven. Voeding, beweging en medicatie worden hierbij ook in acht genomen [17].

3 Methode

3.1 Parameterkeuze en kleurkoppeling

Allereerst is bepaald welke parameters er gebruikt moeten worden in het stoplichtsysteem. Het ligt voor de hand om in beginsel de parameters te kiezen die ook weergegeven worden in het AGP-rapport, aangezien dit bekende gegevens zijn die zorgprofessionals in de praktijk gebruiken [18]. Daarnaast kan het algemene risico en daarmee de regulatie van de glucose-spiegel met één score weergegeven worden. Aangezien er een hoge correlatie is tussen de GRI en HBGI [19], alsmede tussen de Q-Score en HBGI [20], is er in het concept gekozen de bekendere HBGI hiervoor te gebruiken. De risico-categorieën laag, gematigd en hoog kunnen worden gekoppeld aan respectievelijk de kleuren groen, oranje en rood. De combinatie van kleur, HBGI-waarde en naam van de patiënt maakt het huidige risico inzichtelijk. Bij keuze om de HBGI te gebruiken voor de algemene beoordeling ligt de focus op het verminderen van (het aantal) hyperglykemische episodes, maar aangezien er ook ernstige hypoglykemische episodes bij diabetes type 2 patiënten kunnen voorkomen, vooral wanneer zij met insuline behandeld worden, kan ook de weergave van de LBGI zinvol zijn. Het gevaar van hypoglykemie binnen deze groep patiënten wordt namelijk vaak onderschat [21]. Om de betrouwbaarheid van de gemeten parameterwaarden te waarborgen, is het van belang dat het CGM-apparaat minimaal 70% van de tijd actief is, waarbij wordt uitgegaan van 14 dagen [5]. Dit maakt het percentage aan CGM-data ook een essentiële parameter. Tabel 3.1 laat CGM parameterwaarden met bijbehorende kleuren voor het risico zien, die worden gebruikt in het stoplichtsysteem.

De koppeling van de kleuren aan de risicowaarden van de HBGI en LBGI is gebaseerd op de eerdergenoemde definitie van Kovatchev et al. [13]. Het bereik voor de gemiddelde glucose-waarde (mean) is bepaald aan de hand van de correlatie met HbA1c. Een HbA1c-waarde van 53 mmol/L (7%) komt grofweg overeen met een gemiddelde glucose van 8.6 mmol/L [22]. 64 mmol/L (8%) komt overeen met een glucosewaarde van 10.2 mmol/L, wat nagenoeg overeenkomt met de grenswaarde van het groene bereik in het AGP-rapport. Voor de GV is het bepalen

Tabel 3.1: CGM parameters met kleurbereik.

	Groen	Oranje	Rood
HBGI	<4,5	4,5-9,0	>9,0
LBGI	<2,5	2,5-5,0	>5,0
mean (mmol/L)	<8,6	8,6-10,2	>10,2
GV (%CV)	<34	34-40	>40
TIR (%)	>70	50-70	<50
%data	>80	70-80	<70

Groen = goed, Oranje = matig, Rood = slecht.

HBGI = High Blood Glucose Index, LBGI = Low Blood Glucose Index,
mean = gemiddelde bloedglucosewaarde, GV = glucose variabiliteit

als percentage van de variatiecoëfficiënt, TIR = Time In Range,

%data = percentage aan data per tijdsperiode.

van de grenswaarden een grotere uitdaging. Een GV <36% is gestandaardiseerd, maar volgens andere studies komt een waarde <34% overeen met een significant kortere periode in de Time Below Range, hoewel de LBGI een betere indicator zou zijn voor de weergave van het risico van hypoglykemische episodes [23]. Door het verband tussen de GV en de regulatie van de glucosespiegel met behulp van medicatie, zou de streefwaarde gepersonaliseerd moeten worden. Wanneer een patiënt afhankelijk wordt van insuline, is een verhoging van de GV nagenoeg onvermijdelijk [10]. In theorie zou deze bijvoorbeeld zonder medicatie een GV van 30% kunnen hebben, maar met medicatie 40%. Voor deze persoon zou 36% wellicht het best haalbare zijn en 'groen' moeten zijn, ondanks het feit dat dit aanzienlijk hoger is vergeleken met de situatie zonder medicatie.

Als ondergrens voor de TIR is 50% gekozen. Complicaties zoals diabetische neuropathie en retinopathie komen significant vaker voor bij patiënten met een TIR lager dan 50% [24]. Ook geldt dat een TIR van 50% overeenkomt met een HBA1c-waarde van 67 mmol/L [7].

3.2 Bepaling van de tijdsperiode

De aanbevolen tijdsperiode waarover de CGM parameters berekend en beoordeeld zouden moeten worden is 14 dagen, aangezien dit het verloop van de komende drie maanden goed zou kunnen voorspellen. Echter, er zijn ook onderzoeken waarin wordt geconcludeerd dat de data over een langere periode beoordeeld moet worden, vooral wanneer er een nieuwe behandeling wordt ingezet [25]. Ook zou de optimale periode verschillen per parameter, en de frequentie

waarmee het CGM-apparaat de data opslaat. Bij bijvoorbeeld een apparaat dat glucosewaarden opslaat met een interval van 15 minuten is een langere tijdsperiode nodig dan met een interval van 5 minuten [26]. In het geval van de Diameter-studie is dit interval 1 minuut. Om het verschil te zien tussen parameterwaarden voor en na de aanpassing van de behandeling, kan het handig zijn dat de zorgprofessional zelf de gewenste tijdsperiode kan selecteren. De waarden berekend over bijvoorbeeld de afgelopen twee weken kunnen dan vergeleken worden met de waarden van een maand ervoor. De tijdsperiode waarover beoordeeld moet worden hangt af van het beoogde doel, de patiënt en de hoeveelheid verzamelde data.

3.3 Waarschuwingen

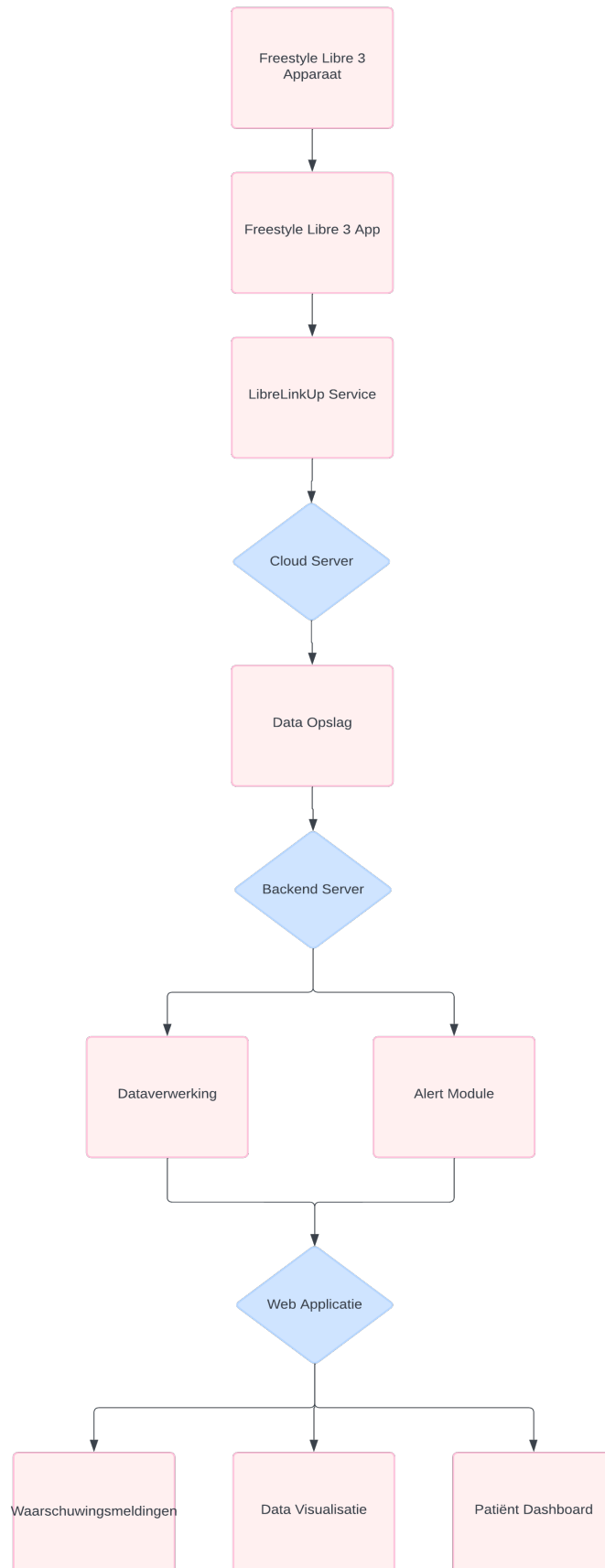
Wanneer de bloedglucoseregulatie van een patiënt dermate verslechtert dat er actie moet worden ondernomen, kan de zorgprofessional gewaarschuwd worden. Een verslechtering van de regulatie is bijvoorbeeld een verlaging van de Time in Range. Voor de TIR geldt dat elke verhoging van 5% klinisch gunstig is [5]. Een verlaging van de TIR van 5% is dan een van de situaties waarvoor een waarschuwing wordt gegeven.

3.4 Dataverwerking en weergave

Voor het berekenen van de parameters en de beoordeling hiervan, moet CGM-data worden verwerkt. Bij de nieuwe FreeStyle Libre 3 wordt er elke minuut een glucosewaarde gemeten. Wanneer deze gekoppeld is met de FreeStyle Libre 3 app, wordt de data meteen opgeslagen in de online tool LibreView. De officiële manier van Abbott is om vanuit LibreView de ruwe data in de vorm van een .csv bestand te downloaden [27]. Dit vereist telkens handmatige downloads, dus real-time is het niet. Mogelijkheden om op andere manieren CGM-waarden te ontvangen en te versturen zijn er wel. In het hoofdstuk *Systeemontwerp* wordt dit verder toegelicht. De data bestaat uit tijdstippen en glucosemetingen, die elke minuut worden opgeslagen. De data wordt ingeladen en verwerkt door een programma geschreven in R. Er kan dan een overzicht gemaakt worden van alle patiënten en de bijbehorende parameters, gekoppeld aan een kleur.

4 Systeemontwerp

In figuur 4.1 is te zien hoe de data van het CGM-apparaat opgeslagen, verwerkt en naar de zorgprofessional verzonden kan worden. De Freestyle Libre 3 wordt gekoppeld met de FreeStyle Libre 3 app door middel van een Bluetooth-verbinding. In deze app is de laatste glucosewaarde samen met historische data van de afgelopen 12 uur te zien. Via de LibreLinkUp service kan deze data verstuurd worden. Hoewel deze service oorspronkelijk bedoeld is om gegevens te delen met vrienden of familie, kan dezelfde koppeling worden gemaakt met een eigen app. LibreLinkUp maakt namelijk gebruik van een API (Application Programming Interface), waarmee gecommuniceerd kan worden om de data op te halen. Zodra de gegevens via de API zijn opgehaald, worden ze opgeslagen in een cloud server. Deze server fungeert als een centrale opslagplaats waar alle patiëntgegevens worden bewaard. De opgeslagen gegevens worden vervolgens door een backend server opgehaald voor verwerking. Tijdens dit proces worden de benodigde parameters berekend. Ook kan er een 'Alert Module' ingebouwd worden die waarschuwingen genereert en verstuurt wanneer dit nodig is, bijvoorbeeld als de TIR van een patiënt significant afneemt. Zo kan de zorgprofessional tijdig ingrijpen en de behandeling aanpassen indien nodig. Het is de bedoeling dat deze 'Alert Module' aanpasbaar is. Dat wil zeggen dat de zorgprofessional in staat moet zijn om de waarschuwingen naar wens in te stellen, zodat deze niet wordt overspoeld met onnodige meldingen.

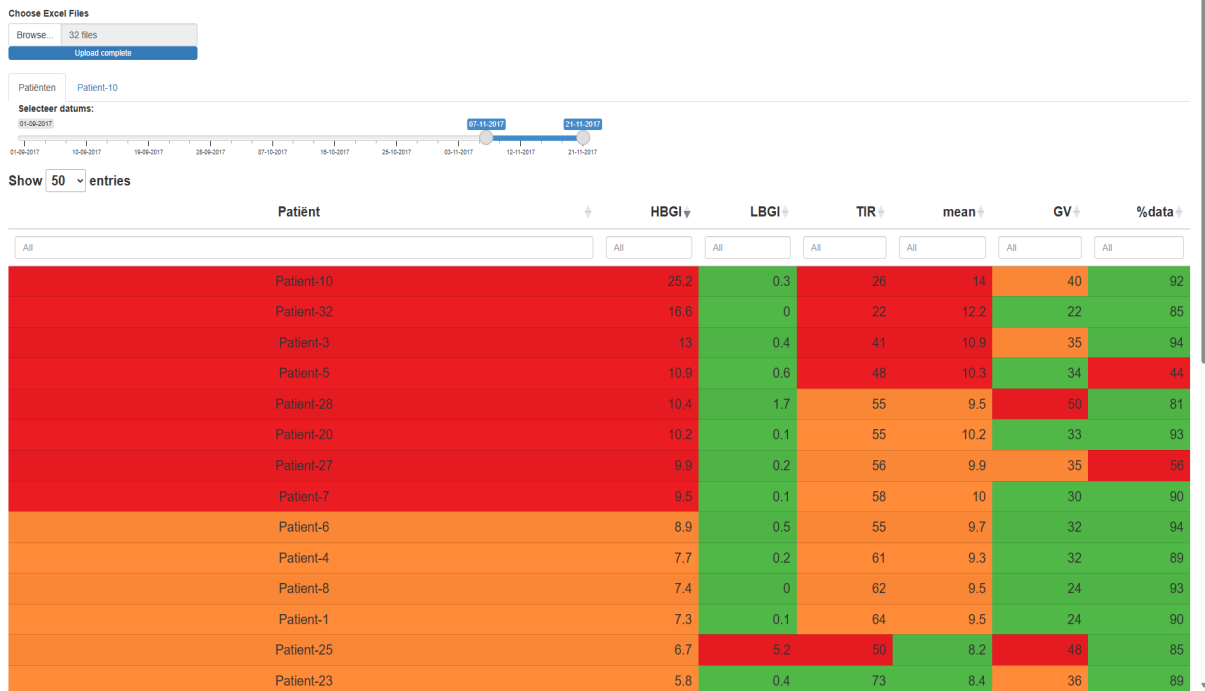


Figuur 4.1: Componenten van het stoplichtsysteem.

5 Systeemconcept

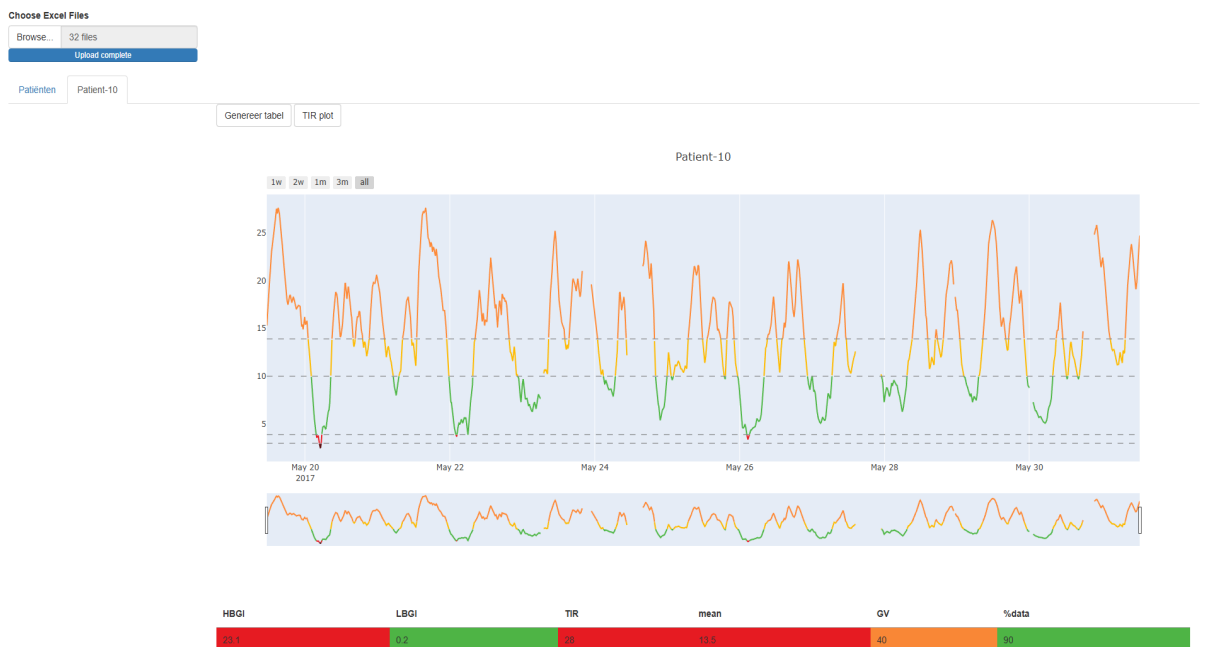
Het uiteindelijke concept is een web-applicatie geschreven in R. De code voor deze app is te vinden in bijlage A en de bijbehorende functies voor het berekenen van de parameters in bijlage B. Deze app biedt een gebruiksvriendelijke interface waar zorgprofessionals de glucosewaarden en bijbehorende gegevens kunnen bekijken. De applicatie maakt gebruik van tabellen en grafieken om de gegevens overzichtelijk en inzichtelijk te maken. In deze app kan de tijdsperiode worden gekozen waarover de parameters berekend worden. De standaard tijdsperiode is de laatste maand waarin de data is verzameld. De namen van alle patiënten worden onder elkaar weergegeven in een lijst, met de bijbehorende parameters erachter. Elk vak met een parameter krijgt een kleur, die weergeeft in welke bereik de waarde zich bevindt. Hierin is rood slecht, oranje matig en groen goed. De lijst is standaard aflopend gesorteerd aan de hand van de HBGI waarde, maar deze kan ook op andere parameters gesorteerd worden. In het overzicht kan er op de naam van de patiënt geklikt worden, waarna er een nieuwe pagina opent met patiënt-specifieke informatie. Hier worden de glucosewaarden weergegeven in een diagram en kunnen parameters berekend over verschillende perioden vergeleken worden. In figuur 5.1 is te zien hoe het patiëntenoverzicht in de app eruit ziet en figuur 5.2 laat de pagina zien die gegenereerd wordt na het klikken op een patiënt.

Patient Data Visualization



Figuur 5.1: Patiëntenoverzicht met parameters en kleuren, gesorteerd op HBGI-waarde.

Patient Data Visualization



Figuur 5.2: CGM-data van een patiënt (Patient-10) weergegeven met bijbehorende parameters.

6 Scenario's en persona's

Om te begrijpen hoe het stoplichtsysteem in de praktijk toegepast kan worden, komen in dit hoofdstuk voorbeelden van scenario's en personen die hierbij betrokken zijn aan bod.

6.1 Scenario 1: Johan de Vries - verpleegkundige

Meneer De Vries is diabetesverpleegkundige, 30 jaar oud, en werkt 5 jaar binnen de diabeteszorg. Hij helpt diabetes type 2 patiënten hun bloedglucosespiegel beter te reguleren door middel van aanpassing in leefstijl. Het aantal patiënten dat bij hem in de kliniek komt stijgt en zijn werkdruk wordt hierdoor verhoogd. De meeste patiënten komen elke drie maanden op controle, maar van een groot deel blijkt dat zij hun bloedglucose over het algemeen zelf goed onder controle hebben. Vaak wordt in deze gevallen slechts een eenvoudige aanpassing in leefstijl zoals toename in beweging en gezondere keuzes qua voeding geadviseerd. Om de werkdruk te verlagen wil hij stoppen met de driemaandelijke controle bij deze groep. Doordat hij real-time CGM gegevens ontvangt, kan hij van elk individu op afstand inzien hoe goed de bloedglucose gereguleerd wordt en of deze patiënt meer aandacht nodig heeft. Hierbij stuit De Vries op een probleem. Hij heeft namelijk moeite om alle patiënten te volgen en mist een inzichtelijk overzicht. Het stoplichtsysteem kan hierbij helpen. Zo kan hij de patiënten in kaart brengen van wie bijvoorbeeld de TIR onvoldoende is. Hij noteert de patiënten met een roodgekleurde TIR en verhoogt de bezoeksfrequentie van deze personen. Aan de patiënten met een groene TIR geeft hij voorlopig digitaal advies.

6.2 Scenario 2: Dirk Jansen - endocrinoloog

Dr. Jansen is 45 jaar oud, en heeft 15 jaar ervaring als internist-endocrinoloog. Samen met de diabetesverpleegkundige en de patiënt stelt hij behandelplannen op. Van de verpleegkundige krijgt hij de vraag om te kijken naar de patiënten die moeite hebben hun glucosespiegel onder controle te houden. In het patiëntenoverzicht in figuur 5.1 is te zien dat er twee patiënten bovenaan staan. Het gaat hierbij om Patient-10 (figuur 5.2) en Patient-32 (figuur 6.1). De

HBGI-waarde van Patient-10 is namelijk 25.2 en die van Patient-32 is 16.6. Het valt dr. Jansen op dat de TIR van Patient-10 wel hoger is. Dus wanneer hij op TIR sorteert, komt Patient-32 juist bovenaan te staan. Ook is er een groot verschil in GV tussen de twee patiënten. Hij besluit de individuele data over een periode van twee weken te analyseren en ziet dat Patient-10 vaak rond 10:00 uur, 14:00 uur, 18:00 uur en soms rond 23:00 uur zeer hoge pieken heeft, terwijl de glucose in de nacht juist vaak erg ver daalt. Op basis hiervan besluit hij de medicijnenlijst van de patiënt te controleren en constateert hij dat de patiënt een hoge dosis middellangwerkende insuline voor de nacht gebruikt. Er wordt besloten deze te vervangen door een mix van zeer kortwerkende en middellangwerkende insuline en de tijd van toediening te verplaatsen naar vlak voor het avondeten. Ook wordt er extra medicatie voorgeschreven voor in de ochtend.



Figuur 6.1: CGM-data van een patiënt (Patient-32) weergegeven met bijbehorende parameters.

Bij Patiënt-32 zijn er voornamelijk vroeg in de ochtend hoge pieken te zien. Ook blijft de glucosewaarde gedurende de dag en nacht hoog. Er zijn geen hypoglykemische episodes. Er wordt besloten een extra tablet voor in de avond voor te schrijven.

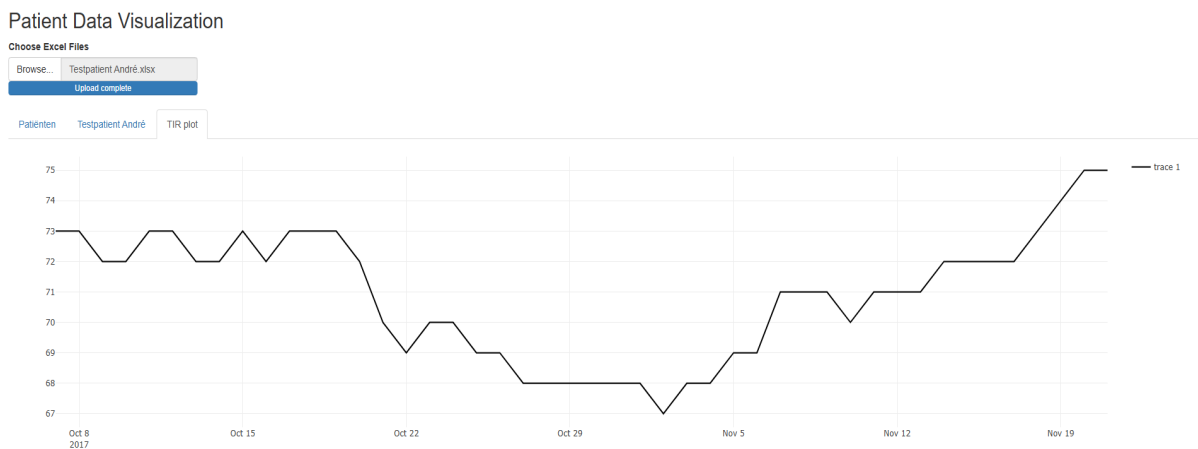
6.3 Scenario 3: André de Jong - patiënt

André is 55 jaar oud en diabetes type 2 patiënt. Op 21-10-2017 had hij een afspraak met de diabetesverpleegkundige. Hem werd verteld dat zijn TIR de afgelopen maand net onder 70% lag. Ook had zijn CGM-sensor twee weken lang een aantal waarden niet geregistreerd, wat ervoor zorgde dat de data maar voor 78% compleet was. Hij probeerde in het vervolg beter te letten op hoe hij zijn sensor aanbracht. Ook zou hij zijn voedingspatroon veranderen. Met behulp van de tabellen in figuur 6.2 is te zien dat zijn TIR en percentage aan data na een maand inderdaad zijn verhoogd. Ook de HBGI en GV zijn verlaagd, maar de gemiddelde glucosewaarde is wel verhoogd. Dit geeft aan dat de hoogte van de pieken is afgenomen en de glucoseregulatie stabiel is geworden.



Figuur 6.2: CGM-data van testpatiënt André. Parameters van twee tijdsperioden worden met elkaar vergeleken.

In figuur 6.3 is de TIR van André over de tijd duidelijker weergegeven. In de eerste week nadat hij op spreekuur kwam nam zijn TIR nog af, maar vanaf 2 november steeg deze. Er is te zien dat de TIR rond 1 november (68%) een aantal dagen 5% lager was dan zijn hoogste waarde (73%). In combinatie met de 'Alert Module' had dit kunnen zorgen voor een waarschuwing bij de zorgprofessional, wanneer dit de ingestelde drempelwaarde zou zijn. Om te voorkomen dat er te snel een waarschuwing gegeven wordt, kan bijvoorbeeld de eis worden gesteld dat de zorgprofessional pas een melding krijgt wanneer de TIR voor een periode van minimaal twee weken 5% lager is.



Figuur 6.3: TIR-waarden van testpatiënt André. De TIR wordt telkens berekend over een periode van één maand vóór de datum op de x-as.

7 Conclusies en aanbevelingen

Een stoplichtsysteem is ontworpen met behulp van bekende CGM-parameters die in de praktijk gebruikt worden. Om de algemene glucoseregulatie te beoordelen is de HBGI een geschikte parameter. Daarnaast vertelt de LBGI iets over het aantal en de hoogte van de waarden binnen het hypoglykemische bereik.

Bij het gebruik van de HBGI en LBGI moet er wel rekening worden gehouden met de CGM-sensor. De FreeStyle Libre 3 sensor meet waarden lager dan 2,2 mmol/L en hoger dan 27,7 mmol/L namelijk niet [28]. De LBGI gedefinieerd is met 1,1 mmol/L als ondergrens en de HBGI met 33,3 mmol/L als bovengrens. Dit betekent dat de berekening van deze parameters moet worden aangepast, of er moet een andere sensor worden gebruikt. Eventueel zou ter vervanging van de HBGI ook de GRI of de Q-score gebruikt kunnen worden, om dit probleem op te lossen.

Verder zijn alle parameterwaarden te koppelen aan de kleuren van het stoplicht, zodat het snel duidelijk is of het een goede, matige of slechte waarde is. Het stoplichtsysteem kan worden ingezet in de praktijk als hulpmiddel voor de zorgprofessional. Het kan namelijk helpen het gezondheidsrisico van de patiënt in te schatten en te beoordelen of deze meer zorg nodig heeft. Het concept dat is ontwikkeld, dient nog verder aangepast en getest te worden. Om het systeem te kunnen gebruiken binnen de PerfectFIT studie, moet het integreerbaar zijn binnen de bestaande applicaties en platforms. Naast het toevoegen van voeding en beweging, moet worden onderzocht hoe alle informatie het beste weergegeven kan worden. Ook de mogelijkheid om extra parameters weer te geven of juist te verbergen, zodat het systeem aan te passen is aan de wensen van de zorgprofessional, moet worden toegevoegd. Voor de 'Alert Module' moet worden onderzocht wanneer, waarvoor en hoe er waarschuwingen gegenereerd moeten worden. Verder zou in een volgend concept de toevoeging van notities, medicijnen en laatste behandeldatum zinvol kunnen zijn. Ook zou de glucose-data in de vorm van AGP-rapporten kunnen worden weergegeven in plaats van in een grafiek, wanneer men op de naam van de patiënt klikt. Vooral voor het analyseren van een periode langer dan een maand zou dit beter zijn, aangezien het genereren van een grafiek met dit aantal datapunten zeer veel tijd in beslag

neemt en er niet meer overzichtelijk uitziet. De AGP-rapporten worden dan gegenereerd voor het geselecteerde tijdsbestek, waarna deze met elkaar vergeleken kunnen worden. Zo sluit het systeem beter aan op de huidige toepassingen in de praktijk. Het systeem kan dan dienen als vervanging van het LibreView dashboard.

Om ervoor te zorgen dat de CGM-data real-time naar het systeem verzonden en verwerkt wordt, moet er extra hardware en software beschikbaar worden gesteld voor de koppeling met de applicatie en het CGM-apparaat. De onderdelen in figuur 4.1 zijn hier een voorbeeld van.

De basis voor een stoplichtsysteem is gegeven, maar de precieze uitwerking en toepassing moet nog worden onderzocht.

Bibliografie

- [1] American Diabetes Association. Diagnosis and Classification of Diabetes Mellitus. *Diabetes Care*. 2013 Dec;37(Supplement_1):S81–S90.
- [2] Xing D, Kollman C, Beck RW, Tamborlane WV, Laffel L, Buckingham BA, et al. Optimal Sampling Intervals to Assess Long-Term Glycemic Control Using Continuous Glucose Monitoring. *Diabetes Technology & Therapeutics*. 2011 Mar;13(3):351–358.
- [3] Goldenberg R, Aroda V, Billings L, Meller Donatsky A, Frederiksen M, Klonoff D, et al. Correlation Between Time in Range and HbA1c in People with Type 2 Diabetes on Basal Insulin: Post Hoc Analysis of the SWITCH PRO Study. *Diabetes therapy : research, treatment and education of diabetes and related disorders*. 2023 May;14(5):915–924.
- [4] Yang L, Xu J, Kang C, Bai Q, Wang X, Du S, et al. Effects of Mobile Phone–Based Telemedicine Management in Patients With Type 2 Diabetes Mellitus: A Randomized Clinical Trial. *The American Journal of the Medical Sciences*. 2022 Mar;363(3):224–231.
- [5] Battelino T, Danne T, Bergenstal RM, Amiel SA, Beck R, Biester T, et al. Clinical Targets for Continuous Glucose Monitoring Data Interpretation: Recommendations From the International Consensus on Time in Range. *Diabetes Care*. 2019 Jun;42(8):1593–1603.
- [6] Lu J, Ma X, Zhou J, Zhang L, Mo Y, Ying L, et al. Association of Time in Range, as Assessed by Continuous Glucose Monitoring, With Diabetic Retinopathy in Type 2 Diabetes. *Diabetes Care*. 2018 Sep;41(11):2370–2376.
- [7] Vigersky RA, McMahon C. The Relationship of Hemoglobin A1C to Time-in-Range in Patients with Diabetes. *Diabetes Technology & Therapeutics*. 2019;21(2):81–85.
- [8] American Diabetes Association Professional Practice Committee. 6. Glycemic Targets: Standards of Medical Care in Diabetes—2022. *Diabetes Care*. 2021 Dec;45(Supplement_1):S83–S96.

- [9] Zoungas S, Chalmers J, Ninomiya T, Li Q, Cooper ME, Colagiuri S, et al. Association of HbA1c levels with vascular complications and death in patients with type 2 diabetes: evidence of glycaemic thresholds. *Diabetologia*. 2011 Dec;55(3):636–643.
- [10] Monnier L, Colette C, Wojtuszczyzn A, Dejager S, Renard E, Molinari N, et al. Toward Defining the Threshold Between Low and High Glucose Variability in Diabetes . *Diabetes Care*. 2016 Dec;40(7):832–838.
- [11] Klonoff DC, Wang J, Rodbard D, Kohn MA, Li C, Liepmann D, et al. A Glycemia Risk Index (GRI) of Hypoglycemia and Hyperglycemia for Continuous Glucose Monitoring Validated by Clinician Ratings. *Journal of Diabetes Science and Technology*. 2023 Sep;17(5):1226–1242.
- [12] Kovatchev BP, Cox DJ, Gonder-Frederick LA, Clarke W. Symmetrization of the Blood Glucose Measurement Scale and Its Applications. *Diabetes Care*. 1997 Nov;20(11):1655–1658.
- [13] Kovatchev BP, Straume M, Cox DJ, Farhy LS. Risk Analysis of Blood Glucose Data: A Quantitative Approach to Optimizing the Control of Insulin Dependent Diabetes. *Journal of Theoretical Medicine*. 2000 Jan;3(1):1–10.
- [14] Kovatchev BP, Otto E, Cox D, Gonder-Frederick L, Clarke W. Evaluation of a New Measure of Blood Glucose Variability in Diabetes. *Diabetes Care*. 2006 Nov;29(11):2433–2438.
- [15] Kovatchev BP. Metrics for glycaemic control - from HbA1c to continuous glucose monitoring. *Nat Rev Endocrinol*. 2017 Jul;13(7):425–436.
- [16] Augstein P, Heinke P, Vogt L, Vogt R, Rackow C, Kohnert KD, et al. Q-Score: development of a new metric for continuous glucose monitoring that enables stratification of antihyperglycaemic therapies. *BMC Endocr Disord*. 2015 May;15(1):22.
- [17] Augstein P, Heinke P, Vogt L, Kohnert KD, Salzsieder E. Patient-Tailored Decision Support System Improves Short- and Long-Term Glycemic Control in Type 2 Diabetes. *Journal of Diabetes Science and Technology*. 2022 Sep;16(5):1159–1166.
- [18] Czupryniak L, Dzida G, Fichna P, Jarosz-Chobot P, Gumprecht J, Klupa T, et al. Ambulatory Glucose Profile (AGP) Report in Daily Care of Patients with Diabetes: Practical Tips and Recommendations. *Diabetes Therapy*. 2022 Mar;13(4):811–821.

- [19] Kim JY, Yoo JH, Kim JH. Comparison of Glycemia Risk Index with Time in Range for Assessing Glycemic Quality. *Diabetes Technology & Therapeutics*. 2023 Nov;25(12):883–892.
- [20] Kohnert KD. Indices for Assessment of the Quality of Glycemic Control and Glucose Dynamics from Continuous Glucose Monitoring. *International Journal of Diabetes and Clinical Research*. 2017 Jun;4(1).
- [21] Heller SR, Peyrot M, Oates SK, Taylor AD. Hypoglycemia in patient with type 2 diabetes treated with insulin: it can happen. *BMJ Open Diabetes Research & Care*. 2020 Jun;8(1):e001194.
- [22] Nathan DM, Kuenen J, Borg R, Zheng H, Schoenfeld D, Heine RJ, et al. Translating the A1C Assay Into Estimated Average Glucose Values. *Diabetes Care*. 2008 Aug;31(8):1473–1478.
- [23] Gómez AM, Muñoz OM, Marin A, Fonseca MC, Rondon M, Gómez MAR, et al. Different Indexes of Glycemic Variability as Identifiers of Patients with Risk of Hypoglycemia in Type 2 Diabetes Mellitus. *Journal of Diabetes Science and Technology*. 2018 Sep;12(5):1007–1015.
- [24] Sheng X, Xiong GH, Yu PF, Liu JP. The Correlation between Time in Range and Diabetic Microvascular Complications Utilizing Information Management Platform. *International Journal of Endocrinology*. 2020;2020(1):8879085.
- [25] Cichosz SL, Jensen MH, Hejlesen O. Optimal Data Collection Period for Continuous Glucose Monitoring to Assess Long-Term Glycemic Control: Revisited. *Journal of Diabetes Science and Technology*. 2022 Jan;17(3):690–695.
- [26] Herrero P, Alalitei A, Reddy M, Georgiou P, Oliver N. Robust Determination of the Optimal Continuous Glucose Monitoring Length of Intervention to Evaluate Long-Term Glycemic Control. *Diabetes Technology & Therapeutics*. 2021 Apr;23(4):314–319.
- [27] Abbott. Can I download my data with the FreeStyle Libre 3 app? [Internet]; 2024. Beschikbaar via: <https://www.support.freestyle.abbott/hc/en-us/articles/14806679954199-Can-I-download-my-data-with-the-FreeStyle-Libre-3-app>. [Geraadpleegd op 17-05-2024].
- [28] Abbott. How accurate is the FreeStyle Libre 14 day system at high and low values? [Internet]; 2023. Beschikbaar via: <https://www.support.freestyle.abbott/hc/en->

us/articles/14772205302295-How-accurate-is-the-FreeStyle-Libre-14-day-system-at-high-and-low-values [Geraadpleegd op 03-07-2024].

A app.R

```
library(shiny)
library(plotly)
library(readxl)
library(dplyr)
library(shinyjs)
library(tidyquant)
library(stringr)
library(lubridate)
library(DT)
options(shiny.maxRequestSize=30*1024^2)
source("functions.R")

# Define UI
ui <- fluidPage(

  titlePanel("Patient Data Visualization"),

  fileInput("files", "Choose Excel Files", multiple = TRUE, accept = c(".xlsx", ".xls")),

  # Tabs for patient plots
  tabsetPanel(
    id = "patient_tabs",

    tabPanel("Patiënten", uiOutput('date_slider'),DTOutput('patient_data'))
  )
)

# Define server logic
server <- function(input, output, session) {

  xvalues <- reactiveVal(NULL) # NULL indicates autorange
```

```

# Reactive values to store patient names and file paths
patient_data <- reactiveValues(colors = NULL, paths = NULL, names = NULL, data_matrix =
  ↪ NULL)
current_patient <- reactiveValues(files = NULL, data = NULL)
filter <- reactiveValues(parameters = NULL)
# Reactive data to store current Tab
Patientslist <- c("Patiënten")
existing_tabs <- reactiveValues(tabs = "Patiënten")

#calc data_matrix function

calc_data_matrix <- function(xvalues,df) {

  HBGI <- calculate_HBGI(xvalues, df)
  LBGI <- calculate_LBGI(xvalues, df)
  TIR <- calculate_TIR(xvalues, df)
  CV <- calculate_CV_mean(xvalues, df)
  mean <- calculate_CV_mean(xvalues, df)
  dataperc <- calculate_dataperc(xvalues, df)

  data_matrix <- (c(HBGI, LBGI, TIR, mean[3:4], CV[1:2], dataperc))

  return(data_matrix)

}

output$date_slider <- renderUI({

  req(input$files)
  fluidPage(
    fluidRow(
      column(width = 6,
        sliderInput(
          "dateSlider",
          width = "100%",
          label = "Selecteer datums:",
          min = as.Date("2017-09-01", format = "%Y-%m-%d"), # Reference start date
          max = as.Date("2017-11-21", format = "%Y-%m-%d"), # Today's date

```

```

        value = c(as.Date("2017-11-07", format = "%Y-%m-%d"), as.Date("2017-11-21",
        ↪ format = "%Y-%m-%d")), # Default selected range
        timeFormat = "%d-%m-%Y" # Format for the date display

    )
  )
)
})

output$patient_data <- DT::renderDataTable({
  req(input$files)
  req(patient_data$data_matrix)

  names <- patient_data$names

  data_matrix <- patient_data$data_matrix

  colors <- data_matrix[seq(2, 12, by = 2), ]
  colors <- t(colors)
  colors <- cbind(colors[,1], colors)
  data <- data_matrix[seq(1, 12, by = 2), ]

  data <- t(data)
  # Check if data_matrix is a matrix
  if (nrow(data) > 1 ) {
    # If it is a matrix, apply as.numeric() to each column and order the data
    data <- apply(data, c(1, 2), as.numeric)
    data <- cbind(names, data.frame(data))
  } else {

    data <- as.numeric(data)
    data <- c(names, data)
    data <- matrix(data, nrow = 1)

  }

  colnames(data) <- c('Patiënt', 'HBGI', 'LBGI', 'TIR', 'mean', 'GV', '%data')

```

```

font.size <- "15pt"
data <- datatable(
  data,
  filter = 'top',
  rownames = FALSE,
  selection = "single",
  options=list(
    dom = 'ltipr',
    columnDefs = list(list(className = 'dt-center', targets = 0)),
    order = list(list(1, 'desc')),
    autoWidth = FALSE,
    pageLength = 50,
    initComplete = htmlwidgets::JS(
      "function(settings, json) {",
      paste0("$(this.api().table().container()).css({'font-size': '", font.size,
      ↵ "});"),
      "}")
    )) %>%
formatStyle(
  c('Patiënt'),
  width = '50%',
  textAlign = 'center'
) %>%
formatStyle(
  c('Patiënt', 'HBGI'), 'HBGI',
  backgroundColor = styleInterval(c(4.5, 9), c('#50B847', '#FF8A37', '#EC1C23'))
) %>%
formatStyle(
  c('LBGI'),
  backgroundColor = styleInterval(c(2.5, 5), c('#50B847', '#FF8A37', '#EC1C23'))
) %>%
formatStyle(
  c('TIR'),
  backgroundColor = styleInterval(c(50, 70), c('#EC1C23', '#FF8A37', '#50B847'))
) %>%
formatStyle(
  c('mean'),
  backgroundColor = styleInterval(c(8.6, 10.2), c('#50B847', '#FF8A37', '#EC1C23'))
) %>%
formatStyle(

```

```

    c('GV'),
    backgroundColor = styleInterval(c(34, 40), c('#50B847', '#FF8A37', '#EC1C23'))
) %>%
formatStyle(
  c('%data'),
  backgroundColor = styleInterval(c(70, 80), c('#EC1C23', '#FF8A37', '#50B847'))
)

return(data)

})

# Extract patient names from file names and store names and paths
observeEvent(list(input$files, input$dateSlider), {
  req(input$dateSlider)
  req(input$files)

  # close all tabs
  for (tab_name in existing_tabs$tabs[-1]) {
    removeTab(inputId = "patient_tabs", target = tab_name)
  }
  existing_tabs$tabs <- "Patient"
  patient_data$names <- tools::file_path_sans_ext(input$files$name)
  patient_data$paths <- input$files$datapath

  patient_data$data_matrix <- sapply(patient_data$names, function(name) {
    df <- read_excel(patient_data$paths[[which(patient_data$names == name)]])

    colnames(df) <- c("tGluc", "gluc")

    if (!is.null(input$dateSlider)){

      xvalues(c(input$dateSlider[1], input$dateSlider[2]))
    }

    else {xvalues(c(df$tGluc[1], tail(df$tGluc, n = 1)))
    }
  })

```

```

data_matrix <- calc_data_matrix(xvalues(),df)

return(data_matrix)

})

})

# Load data for selected patient and plot

row <- observeEvent(input$patient_data_rows_selected,{
  req(input$patient_data_rows_selected)
  row <- input$patient_data_rows_selected

  name <- patient_data$names[row]

  # remove all tabs
  for (tab_name in existing_tabs$tabs[-1]) {
    removeTab(inputId = "patient_tabs", target = tab_name)
  }

  existing_tabs$tabs <- c(existing_tabs$tabs, name)

  #delete all generated tables
  saved_tables$tables <- NULL
  saved_tables$date_ranges <- NULL

  insertTab(inputId = "patient_tabs",
    tabPanel(title = name,
      fluidPage(
        fluidRow(
          column(width = 2,
            uiOutput("saved_tables")
          ),
          column(width = 10,
            actionButton("save_table", "Genereer tabel"),
            actionButton("TIR_plot", "TIR plot"),
            plotlyOutput("glucosePlot", height = "100%", width = "100%"),

```

```

        DTOutput("current_table")
    )
)
)))

```

```

output$glucosePlot <- renderPlotly({

  glucose <- read_excel(patient_data$paths[row])

  current_patient$files <- glucose

  colnames(glucose) <- c("tGluc", "gluc")

  if (length(glucose$tGluc) > 129600){

    gluc <- tail(glucose$gluc, 129600)
    tGluc <- tail(glucose$tGluc, 129600)

  } else{

    gluc <- glucose$gluc
    tGluc <- glucose$tGluc
  }

  ranges <- list(
    gluc < 3.0,
    gluc >= 3.0 & gluc < 3.9,
    gluc >= 3.9 & gluc <= 10.0,
    gluc > 10.0 & gluc <= 13.9,
    gluc > 13.9
  )

  colors <- c('#5B1213', '#EC1C23', '#50B847', '#FFBB00', '#FF8A37')

  p <- plot_ly(glucose, type = 'scatter', mode = 'lines', height = 700)

  for (i in 1:length(ranges)) {
    gluc1 <- ifelse(ranges[[i]], gluc, NA)
    p <- add_trace(p, x = tGluc, y = gluc1, line = list(color = colors[i]))
  }
}
)

```



```

}

date1 <- as.Date(input$dateSlider[1])
date2 <- as.Date(input$dateSlider[2])
p <- layout(p, showlegend = F, title=name,
            xaxis = list(range = c(tGluc[1],tail(tGluc,n=1)),
                          rangeslider = list(visible = T),
                          rangeselector = list(
                            buttons = list(
                              list(count = 7, label = "1w", step = "day", stepmode =
                                ↪ "backward"),
                              list(count = 14, label = "2w", step = "day", stepmode =
                                ↪ "backward"),
                              list(count = 1, label = "1m", step = "month", stepmode =
                                ↪ "backward"),
                              list(count = 3, label = "3m", step = "month", stepmode =
                                ↪ "backward"),
                              #list(count = 6, label = "6m", step = "month", stepmode =
                                ↪ "backward"),
                              #list(count = 1, label = "YTD", step = "year", stepmode =
                                ↪ "todate"),
                              #list(count = 1, label = "1y", step = "year", stepmode =
                                ↪ "backward"),
                              list(step = "all")
                            )
                          )))

hline <- function( y = 0, color = '#7A7A7A'){
  list(
    type = "line",
    x0 = 0,
    x1 = 1,
    xref = "paper",
    y0 = y,
    y1 = y,
    line = list(color = color, width = 1, dash = "dash")
  )
}

p <- p %>%
  layout(shapes = list(hline(3.0), hline(3.9), hline(10.0), hline(13.9)),

```

```

    xaxis = list(zerolinecolor = '#fff',
                zerolinewidth = 2,
                gridcolor = '#fff'),
    yaxis = list(zerolinecolor = '#fff',
                zerolinewidth = 2,
                showgrid = F),
    plot_bgcolor = '#e5ecf6', margin = 0.1)

  #xvalues(c(tGluc[1], tail(tGluc, n=1)))
  return(p)
})

#switch to tab

updateTabsetPanel(session, inputId = "patient_tabs", selected = name)
})

#observe selected range

observe({

  req(current_patient$files)
  d <- event_data("plotly_relayout")
  if (isTRUE(d$xaxis.autorange)){

  } else if (isTruthy(d$xaxis.range)) { # As triggered by the rangeslider
    xvalues(c(d$xaxis.range[1], d$xaxis.range[2]))
  } else if (isTruthy(d[["xaxis.range[0]"]])) { # As triggered by the plot
    xvalues(c(d[["xaxis.range[0]"]], d[["xaxis.range[1]"]]))
  }

  #calc current patient data vector

  df <- current_patient$files
  colnames(df) <- c("tGluc", "gluc")

  xvalues <- xvalues()

```

```

data_vector <- calc_data_matrix(xvalues,df)

colors <- data_vector[seq(2, 12, by = 2)]
colors <- cbind(colors[1], colors)
colors <- t(colors)
data <- data_vector[seq(1, 12, by = 2)]
data <- t(data)

colnames(data) <- c('HBGI', 'LBGI', 'TIR', 'mean', 'GV', '%data')

current_patient$data <- data

proxy1 <- DT::dataTableProxy('current_table')
DT::replaceData(proxy1, current_patient$data)

})

initial_tabledata <- function(){

req(current_patient$files)

df <- current_patient$files
colnames(df) <- c("tGluc", "gluc")

xvalues <- c(df$tGluc[1], tail(df$tGluc, n=1))

data_vector <- calc_data_matrix(xvalues,df)

colors <- data_vector[seq(2, 12, by = 2)]
colors <- cbind(colors[1], colors)
colors <- t(colors)
data <- data_vector[seq(1, 12, by = 2)]
data <- t(data)

colnames(data) <- c('HBGI', 'LBGI', 'TIR', 'mean', 'GV', '%data')

return(data)
}

saved_tables <- reactiveValues(tables = list(), date_ranges = list())

```

```
#render table with plot
```

```
output$current_table <- DT::renderDataTable({

  data <- datatable(
    initial_tabledata(),
    rownames = FALSE,
    selection = "none",
    options = list(
      autoWidth = FALSE,
      paging = FALSE,      # Disable paging
      searching = FALSE,   # Disable search
      ordering = FALSE,    # Disable ordering
      info = FALSE,        # Disable info
      select = NULL,       # Disable row selection
      columnDefs = list(
        list(targets = "_all", searchable = FALSE) # Disable individual column search
      )
    )
  )%>%

  formatStyle(
    c('HBGI'),
    backgroundColor = styleInterval(c(4.5, 9), c('#50B847', '#FF8A37', '#EC1C23'))
  ) %>%

  formatStyle(
    c('LBGI'),
    backgroundColor = styleInterval(c(2.5, 5), c('#50B847', '#FF8A37', '#EC1C23'))
  ) %>%

  formatStyle(
    c('TIR'),
    backgroundColor = styleInterval(c(50, 70), c('#EC1C23', '#FF8A37', '#50B847'))
  ) %>%

  formatStyle(
    c('mean'),
    backgroundColor = styleInterval(c(8.6, 10.2), c('#50B847', '#FF8A37', '#EC1C23'))
  ) %>%

  formatStyle(
    c('GV'),
    backgroundColor = styleInterval(c(34, 40), c('#50B847', '#FF8A37', '#EC1C23'))
  )
})
```

```

) %>%
formatStyle(
  c('%data'),
  backgroundColor = styleInterval(c(70, 80), c('#EC1C23', '#FF8A37', '#50B847'))
)

return(data)

})

observeEvent(input$TIR_plot, {

  removeTab(inputId = "patient_tabs", target = "TIR_plot")
existing_tabs$tabs <- c(existing_tabs$tabs[-3])

insertTab(inputId = "patient_tabs",
  tabPanel(title = "TIR plot",
    fluidPage(plotlyOutput("TIRplot", height = "100%", width = "100%")))

existing_tabs$tabs <- c(existing_tabs$tabs, "TIR plot")

glucose <- current_patient$files

# Convert tGluc to dates
Dates <- unique(as.Date(glucose$tGluc))

TIR <- sapply(1:(length(Dates) - 29), function(i) {
  dates <- c(Dates[i], Dates[i + 29])
  calculate_TIR(dates, glucose)
})

TIR_values <- TIR[seq(1, length(TIR), by = 2)]

if (length(Dates[-1:-29])==length(TIR_values)){
  data <- tibble(
    tGluc = Dates[-1:-29],
    gluc = TIR_values)

```

```

output$TIRplot <- renderPlotly({

  p <- plot_ly(data, type = 'scatter', mode = 'lines')
  p <- add_trace(p, x = data$tGluc, y = data$gluc, line = list(color = "black"))

  return(p)

})

}

updateTabsetPanel(session, inputId = "patient_tabs", selected = "TIR plot")

})

observeEvent(input$save_table, {
  table_id <- paste0("table_", length(saved_tables$tables) + 1)
  saved_tables$tables[[table_id]] <- current_patient$data
  saved_tables$date_ranges[[table_id]] <- paste0(format(as.POSIXct(xvalues()[1], format =
  ↪ "%Y-%m-%d %H:%M"), "%d-%m-%Y %H:%M"), " - ", format(as.POSIXct(xvalues()[2], format
  ↪ = "%Y-%m-%d %H:%M"), "%d-%m-%Y %H:%M"))
})

output$saved_tables <- renderUI({
  lapply(names(saved_tables$tables), function(table_id) {
    checkbox_id <- paste0("checkbox_", sub("table_", "", table_id))
    fluidRow(
      column(width = 1, checkboxInput(checkbox_id, label = "", value = TRUE)),
      column(width = 11, div(id = paste0("div_", table_id),
      ↪ dataTableOutput(paste0("saved_", table_id))))
    )
  })
})

observe({
  lapply(names(saved_tables$tables), function(table_id) {
    checkbox_id <- paste0("checkbox_", sub("table_", "", table_id))

```

```

output[[paste0("saved_", table_id)]] <- renderDataTable({
  if (!is.null(input[[checkbox_id]]) && input[[checkbox_id]]) {
    datatable(
      saved_tables$tables[[table_id]],
      caption = saved_tables$date_ranges[[table_id]],
      rownames = FALSE,
      selection = "none",
      options = list(
        autoWidth = FALSE,
        paging = FALSE,
        searching = FALSE,
        ordering = FALSE,
        info = FALSE,
        select = NULL,
        columnDefs = list(
          list(targets = "_all", searchable = FALSE)
        )
      )
    ) %>%
    formatStyle(
      c('HBGI'),
      backgroundColor = styleInterval(c(4.5, 9), c('#50B847', '#FF8A37', '#EC1C23'))
    ) %>%
    formatStyle(
      c('LBGI'),
      backgroundColor = styleInterval(c(2.5, 5), c('#50B847', '#FF8A37', '#EC1C23'))
    ) %>%
    formatStyle(
      c('TIR'),
      backgroundColor = styleInterval(c(50, 70), c('#EC1C23', '#FF8A37', '#50B847'))
    ) %>%
    formatStyle(
      c('mean'),
      backgroundColor = styleInterval(c(8.6, 10.2), c('#50B847',
        ↪ '#FF8A37', '#EC1C23'))
    ) %>%
    formatStyle(
      c('GV'),
      backgroundColor = styleInterval(c(34, 40), c('#50B847', '#FF8A37', '#EC1C23'))
    ) %>%

```

```

    formatStyle(
      c('%data'),
      backgroundColor = styleInterval(c(70, 80), c('#EC1C23', '#FF8A37', '#50B847'))
    )
  }
})
})
})

observe({
  lapply(names(saved_tables$tables), function(table_id) {
    checkbox_id <- paste0("checkbox_", sub("table_", "", table_id))
    observeEvent(input[[checkbox_id]], {
      if (is.null(input[[checkbox_id]]) || !input[[checkbox_id]]) {
        saved_tables$tables[[table_id]] <- NULL
        saved_tables$date_ranges[[table_id]] <- NULL
        removeUI(selector = paste0("#div_", table_id))
      }
    }, ignoreInit = TRUE)
  })
})
}

# Run the application
shinyApp(ui = ui, server = server)

```


B functions.R

```
xtogluc <-function(xvalues, patient_data) {

  glucose <- patient_data
  tGluc <- glucose$tGluc
  gluc <- glucose$gluc

  time1 <- as.POSIXct(xvalues[1], format = "%Y-%m-%d %H:%M")
  time2 <- as.POSIXct(xvalues[2], format = "%Y-%m-%d %H:%M")

  idx1 <- match(time1,tGluc, nomatch = 0)
  idx2 <- match(time2,tGluc, nomatch = 0)
  if (idx1 == 0){idx1 <- 1}
  if (idx2 == 0){idx2 <- length(tGluc)}
  selectedgluc <- gluc[idx1:idx2]

  return(selectedgluc)

}

calculate_TIR <- function(xvalues,patient_data) {

  selectedgluc <- xtogluc(xvalues, patient_data)
  selectedgluc <- na.omit(selectedgluc)
  numinrange <- sum(selectedgluc >= 3.9 & selectedgluc <= 10.0)
  TIR <- round(numinrange/length(selectedgluc)*100)

  if (TIR < 50) {
    TIR_color <- '#EC1C23' # Red
  } else if (TIR >= 50 && TIR <= 70) {
    TIR_color <- '#FF8A37' # Orange
  } else {
```

```

    TIR_color <- '#50B847' # Green
  }

  return(c(TIR, TIR_color))
}

calculate_HBGI <- function(xvalues,glucose_data) {

  selectedgluc <- xtogluc(xvalues,glucose_data)
  # Define the function f(BG)
  f <- function(BG) {
    (1.794 * ((log(BG)) ^ (1.026) - 1.861))
  }

  # Calculate HBGI for the patient
  # Filter out NAN values
  glucose <- selectedgluc[!is.na(selectedgluc)]

  # Calculate f(BG) for each blood glucose value
  f_BG <- f(glucose)

  # Replace negative values of f(BG) with 0
  f_BG[f_BG <= 0] <- 0

  # Define the function r(BG)
  r_BG <- 10 * f_BG^2

  # Calculate the HBGI
  HBGI <- sum(r_BG) / length(r_BG)

  # Determine box color based on HBGI value
  if (HBGI < 4.5) {
    HBGI_color <- '#50B847' # Green
  } else if (HBGI >= 4.5 && HBGI <= 9) {
    HBGI_color <- '#FF8A37' # Orange
  } else {
    HBGI_color <- '#EC1C23' # Red
  }
  HBGI <- c(round(HBGI, digits = 1), HBGI_color)
}

```

```

    return(HBGI)
}

calculate_LBGI <- function(xvalues,glucose_data) {

  selectedgluc <- xtogluc(xvalues,glucose_data)
  # Define the function f(BG)
  f <- function(BG) {
    (1.794 * ((log(BG)) ^ (1.026) - 1.861))
  }

  # Calculate HBGI for the patient
  # Filter out NAN values
  glucose <- selectedgluc[!is.na(selectedgluc)]

  # Calculate f(BG) for each blood glucose value
  f_BG <- f(glucose)

  # Replace negative values of f(BG) with 0
  f_BG[f_BG >= 0] <- 0

  # Define the function r(BG)
  r_BG <- 10 * f_BG^2

  # Calculate the HBGI
  LBGI <- sum(r_BG) / length(r_BG)

  # Determine box color based on HBGI value
  if (LBGI < 2.5) {
    LBGI_color <- '#50B847' # Green
  } else if (LBGI >= 2.5 && LBGI <= 5) {
    LBGI_color <- '#FF8A37' # Orange
  } else {
    LBGI_color <- '#EC1C23' # Red
  }
  LBGI <- c(round(LBGI, digits = 1), LBGI_color)

  return(LBGI)
}

```

```

calculate_CV_mean <- function(xvalues, glucose_data) {

  selectedgluc <- xtogluc(xvalues,glucose_data)

  non_nan_selectedgluc <- !is.na(selectedgluc)
  glucose <- selectedgluc[non_nan_selectedgluc]

  # Calculate mean and standard deviation of glucose values
  glucose_mean <- mean(glucose)
  glucose_std <- sd(glucose)

  if (is.na(glucose_std)) {glucose_std = 0}

  CV <- (glucose_std / glucose_mean) * 100

  if (CV > 40) {
    color_CV <- '#EC1C23' # Red
  } else if (CV >= 34 && CV <= 40) {
    color_CV <- '#FF8A37' # Orange
  } else {
    color_CV<- '#50B847' # Green
  }

  if (glucose_mean > 10.2) {
    color_mean <- '#EC1C23' # Red
  } else if (glucose_mean >= 8.6 && glucose_mean <= 10.2) {
    color_mean <- '#FF8A37' # Orange
  } else {
    color_mean <- '#50B847' # Green
  }
  return(c(round(CV, digits = 0), color_CV, round(glucose_mean, digits = 1), color_mean))
}

```

```

calculate_dataperc <- function(xvalues, glucose_data) {

  selectedgluc <- xtogluc(xvalues,glucose_data)

  total_values <- length(selectedgluc)
  non_nan_selectedgluc <- !is.na(selectedgluc)

```

```
total_non_nan_values <- sum(non_nan_selectedgluc)

dataperc <- total_non_nan_values/total_values*100

if (dataperc <=70) {
  color_data <- '#EC1C23' # Red
} else if (dataperc > 70 && dataperc <= 80) {
  color_data <- '#FF8A37' # Orange
} else {
  color_data<- '#50B847' # Green
}
return(c(round(dataperc, digits = 0), color_data))
```