#### Sistem za upravljanje glukozom u krvi kod osoba sa dijabetesom tip 1 1. Spisak članova tima Balša Bulatović 2. Motivacija

insulina, dodatni unos hrane, prilagođavanje aktivnosti). Komercijalna rešenja su često zatvorena i teško

## Dijabetes tip 1 zahteva kontinuirano praćenje glukoze i brze, kontekstualne odluke (korektivna doza brzog

prilagodljiva. Rule-based pristup (Drools) obezbeđuje transparentna, izmenjiva pravila i preciznu personalizaciju (doba dana, intenzitet aktivnosti, senzitivnost na insulin, nutritivni profil hrane).

3. Opis problema

Sistem u intervalima od X minuta prima merenja CGM-a, beleži obroke i doze insulina, te predlaže konkretne akcije: tačne namirnice/porcije ili korektivne doze, uz uvažavanje ugljenih hidrata (UH), masti i glikemijskog indeksa (GI). Potrebno je rešiti: 1. pravovremeno otkrivanje rizika (hipo/hiper, trendovi),

#### 2. preporuku tačne hrane (iz SQL baze), 3. izbegavanje "stack-ovanja" insulina (IOB),

4. lako prilagođavanje logike pravilima i template-ima (bez rekompajliranja). 4. Metodologija rada 4.1 Ulazi u sistem

ActivationEvent (umesto ActivityEvent): type, intensity (LOW/MED/HIGH/NONE), startTime,

#### duration\_min • UserProfile: carbRatio (g/U), correctionFactor (mmol/L po U), targetRange (min, max), insulinActionHours, time-of-day factors SystemState (derivisano): iob (U)

4.2 Izlazi iz sistema Predlozi akcija: Korektivna doza (npr. "uzeti 1.5 U insulina", izračunato iz pravila/servisa). • Lista predloženih namirnica iz SQL baze (ne tekstualni gram-iznosi): npr. "jabuka 180 g", "ovsene 40 g + jogurt 100 ml", "pirinčani keks 2×10 g".

•  $IOB(t) = \Sigma$  remaining\_i(t) za sve InsulinShotEvent sa ts  $\leq$  t.

GlucoseMeasurement( value >= 4.0, value < 6.0 ) &&

GlucoseMeasurement( value >= 4.0, value < 4.8 ) &&

insertLogical( new FoodConstraint(25, 40, 10, "MED") );

insertLogical( new AvoidProfile( giClass="HIGH", fatMin=15 ) );

and GlucoseMeasurement( this after[10m] \$g1, value > \$g1 + 1.5 )

// dalje obrađuje servis: predlog doze + eventualno FoodConstraint za low GI/masti

Pored trenutnog merenja glukoze, sistem mora da uvaži i **nedavni istorijat ishrane**. Zato se koristi

Food( \$gi: glycemicIndex, \$f: fat\_g, \$c: carbs\_g ) over window:time(4h),

accumulate kako bi se na osnovu unosa hrane u prethodnom periodu izračunali agregirani indikatori:

Ako je \$sumCarb u poslednja 4 sata > 150 g, sistem pooštrava korektivni faktor (manje toleriše dodatne

Ako je \$avgGi > 65 i \$avgFat > 20, preporučene namirnice moraju imati niži GI i manje masti.

Ako je \$maxCarb (jedan obrok) > 80 g, sistem detektuje "težak obrok" i aktivira dodatno praćenje za

and GlucoseMeasurement( this after[10m] value > \$g1 + 3.0 )

D) Blizu donje granice + predstoji zahtevnija aktivnost:

Trend( direction == UP, strength >= 2 ) &&

GlucoseMeasurement( value >= 9.0 )

GlucoseMeasurement( \$g1: value )

insert( new Trend(UP, strength:2) );

Trend( direction == UP, strength >= 2 ) &&

not InsulinShotEvent( this after[0m,2h] )

insertLogical( new FoodConstraint(10, 20, 8, "LOW") );

4.3.3 Accumulate (dinamičko prilagođavanje pravila)

GlucoseMeasurement( value > 10 ) &&

insert( new CorrectionNeeded() );

3. Akcije nad hranom i/ili insulinom

: average(\$gi),

kasni porast glukoze (spajanje sa CEP).

A) Brzi pad (rizik hipo) – detekcija i reakcija:

\$a : GlucoseMeasurement( \$v1: value )

4.3.4 CEP (Complex Event Processing)

\$avgFat : average(\$f),

\$sumCarb : sum(\$c), \$maxCarb : max(\$c)

ugljene hidrate).

CorrectionNeeded()

eval( iob() < 1.0 ) &&

insertLogical( new FoodConstraint(20, 35, 12, "LOW") );

( ActivationEvent( intensity == LOW ) || ActivationEvent( intensity == NONE ) )

ActivationEvent( intensity in (MED, HIGH), startTime after now, duration\_min >= 30 )

GlucoseMeasurement: value (mmol/L), timestamp

• InsulinShotEvent: units, timestamp

Food: name, carbs\_g, fat\_g, glycemicIndex, portionSize\_g

# 4.3 Baza znanja

4.3.0 Računanje IOB (iz InsulinShotEvent)

IOB(t) = suma preostalih jedinica brzog insulina u trenutku t od svih prethodnih InsulinShotEvent. Jednostavan eksponencijalni model: • Za ubrizgavanje u u vreme ts: remaining(t) = u \* exp(-k \*  $\Delta h$ ), gde je  $\Delta h$  = (t - ts) u satima,  $k \approx 3 / insulinActionHours$ .

#### 4.3.1 Pravila A) Hipoglikemija → traži brze šećere iz baze (bez tekstualne poruke): when

GlucoseMeasurement( value < 3.9 )</pre> then // ciljaj 15-18 g brzih UH, minimalne masti, HIGH/FAST GI insertLogical( new FoodConstraint(15, 18, 5, "HIGH") ); B) Hiperglikemija bez nedavne korekcije (sprečavanje stack-a):

#### when GlucoseMeasurement( \$g : value > 10 ) && not InsulinShotEvent( this after[0m,2h] \$g ) then insert( new CorrectionNeeded() );

C) Postprandijalna modulacija (sporiji porast → low GI, umerene masti): when then

## when then E) Visoka glukoza + trend ↑ (zabrana profila hrane):

when then 4.3.2 Forward chaining (3+ nivoa) Scenario: kasni porast zbog masti 1. Detekcija trenda when

# 2. Uslov za korekciju (IOB nizak, nema skoro date doze)

then

when

then

then

accumulate(

\$avgGi

)

when

when

and not InsulinShotEvent( this before[0m] \$b ) then insert( new RapidDropEvent(delta: (\$v1 - value(\$b))) ); end when RapidDropEvent( delta >= 1.5 ) && GlucoseMeasurement( value < 4.5 )</pre>

B) "Fat-induced late spike" (kasni porast zbog masti):

MealEvent( totalFat\_g >= 25 ) over window:time(3h) and GlucoseMeasurement( value >= 6.0, value <= 8.0 )

and not Trend( direction == UP, strength >= 2 )

and \$b : GlucoseMeasurement( this after[5m] \$a, value < \$v1 - 1.5 )

insertLogical( new FoodConstraint(15, 20, 5, "HIGH") ); // brzi UH iz baze

insert( new HighFatWatch( durationMin:180 ) ); end when HighFatWatch() and GlucoseMeasurement( \$g1: value ) and GlucoseMeasurement( this after[30m] \$g1, value >= \$g1 + 2.0 ) insertLogical( new FoodConstraint(10, 20, 8, "LOW") ); // mini-užina sa low GI/mastima

#### 4.3.5 Template — FoodRecommendationByContext → FoodConstraint (ulančavanje) Template ne emituje poruku; generiše **FoodConstraint** iz "konteksta" (glukoza + aktivnost). **CSV** primer:

glucoseMin,glucoseMax,activitySet,durationMin,carbsMin,carbsMax,fatMax,giClass 4.0,6.0, NONE | LOW, 30, 20, 35, 12, LOW

4.0,4.8,MED|HIGH,30,25,40,10,MED 7.0,10.0,NONE|LOW,0,10,20,12,LOW

template foodRecommendationByContext

glucoseMin glucoseMax activitySet durationMin carbsMin carbsMax fatMax giClass

GlucoseMeasurement( value >= @{glucoseMin}, value < @{glucoseMax} )</pre>

Template (.drt):

when

28

then

end

));

#### ( ( @{durationMin} == 0 && ( ActivationEvent( intensity == NONE ) || not ActivationEvent() ) ) Ш ( ActivationEvent( ( @{activitySet} ).contains(intensity), duration\_min >= @{durationMin} ) ) )

insertLogical( new FoodConstraint(

@{carbsMin}, @{carbsMax}, @{fatMax}, "@{giClass}"

4.3.6 Ulančavanje na template: izbor sve hrane iz SQL baze Varijanta A — SQL fetch (preporučeno): rule "Fetch foods for constraint" no-loop true salience 5 when

#### \$fc : FoodConstraint( \$cMin : carbsMin, \$cMax : carbsMax, \$fatMax : fatMax, \$gi : giClass ) java.util.List<Food> foods = foodService.findByConstraints(\$cMin, \$cMax, \$fatMax, \$gi); for (Food f : foods) { insert( new SuggestedFood(f) ); }

Varijanta B — in-memory simulacija:

no-loop true salience 5

when

then

giClass )

rule "Match foods for constraint (in-memory)"

fat\_g <= \$fatMax )</pre>

\$f : Food( glycemicIndex == \$gi,

insert( new SuggestedFood(\$f) );

\$fc : FoodConstraint( \$cMin : carbsMin, \$cMax : carbsMax, \$fatMax : fatMax, \$gi :

carbs\_g >= \$cMin, carbs\_g <= \$cMax,</pre>

end