

Računalni sustav namijenjen praćenju pokazatelja zdravstvenog stanja oboljelih od dijabetesa

Bartolić, Luka

Master's thesis / Diplomski rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:329565>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-05-14**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Sveučilišni studij

**RAČUNALNI SUSTAV NAMIJENJEN PRAĆENJU
POKAZATELJA ZDRAVSTVENOG STANJA
OBOLJELIH OD DIJABETESA**

Diplomski rad

Luka Bartolić

Osijek, 2018.

SADRŽAJ

1. UVOD	2
2. DIJABETES I PROBLEMI KOJE DONOSI.....	4
2.1. Problemi s kojima se nose oboljeli.....	6
2.2. Dijagnostički i terapijski postupci.....	9
2.2.1. Faktori rizika i dijagnoza.....	9
2.2.2. Prevencija i liječenje	11
2.3. Važni pokazatelji zdravstvenog stanja i njihovo praćenje	15
2.4. Provedena istraživanja i postojeća programska rješenja	22
2.4.1. Few Touch Application.....	23
2.4.3. Ostala rješenja u upotrebi.....	26
3. PROGRAMSKO RJEŠENJE – SUSTAV ZA PRAĆENJE ZDRAVSTVENOG STANJA OBOLJELIH OD DIJABETESA	30
3.1. Opis platformi, alata i tehnologija.....	31
3.1.1. LAMP stog	31
3.1.2. Laravel MVC razvojni okvir	32
3.1.3. JavaScript, HTML, CSS i Sass.....	34
3.1.4. Fitbit i OAuth 2.0	38
3.1.5. Android.....	42
3.3. Razvoj i način rada sustava za praćenje zdravstvenog stanja oboljelih od dijabetesa	49
3.3.1. Baza podataka i modeli	49
3.3.2. Sinkronizacija s Fitbit uređajem.....	52
3.3.3. Sinkronizacija s glukometrom.....	55
3.3.4. Web aplikacija.....	56
3.3.5. Android mobilna aplikacija.....	76
4. ZAKLJUČAK	87
5. LITERATURA.....	88
SAŽETAK.....	92
ABSTRACT	93
ŽIVOTOPIS	94
PRILOZI.....	95

1. UVOD

Tema ovog rada odnosi se na mogućnost primjene suvremenih informacijskih i komunikacijskih tehnologija (ICT, engl. *Information and communication technologies*) u olakšavanju svakodnevnog života, praćenju i nadzoru zdravstvenog stanja osoba oboljelih od šećerne bolesti ili dijabetesa. U praktičnom dijelu rada ostvareno je programsko rješenje koje prikazuje korist od povezivanja mobilnih, komunikacijskih i informacijskih tehnologija te njihovu primjenu i dobrobit u zdravstvu i upravljanju dijabetesom. Razvijeni sustav namijenjen praćenju zdravstvenog stanja oboljelih od dijabetesa sastoji se od: (1) prenosivog senzorskog uređaja (FitBit Charge HR) za praćenje fizičke aktivnosti i nekih fizioloških parametara, (2) Contour Next One Bluetooth glukometara za praćenje razine glukoze u krvi, (3) Android mobilne aplikacije za automatizirano prikupljanje podataka, dvosmjernu komunikaciju s poslužiteljem te pružanje pristupačnog korisničkog sučelja pacijentu i (4) web rješenja ili e-platforme koja s jedne strane služi kao poslužitelj (engl. *server*) koji povezuje cijeli sustav u cjelinu te za pohranu i sinkronizaciju podataka, njihovu obradu, analizu, sigurnost i distribuciju, a s druge strane kao aplikacija koja korisnicima pruža korisničko sučelje.

Kronične bolesti poput dijabetesa ogroman su teret zdravstvenih institucija i javnog zdravstva. Velika rasprostranjenost dijabetesa u Hrvatskoj i na globalnoj razini, komplikacije koje donosi te ekonomska opterećenja, potiču na intenzivna medicinska istraživanja, a sve više se pokušava poboljšati zdravstvena zaštita pacijenata iskorištavanjem naprednih mogućnosti spomenutih tehnologija. Pacijenti oboljeli od kroničnih bolesti moraju svakodnevno pratiti i bilježiti brojne zdravstvene pokazatelje i fiziološke parametre vlastitog zdravlja poput razine šećera u krvi i rada srca. U sadašnjem zdravstvenom sustavu, uglavnom se ovi bitni podaci bilježe na papiru gdje nastaju problemi u praćenju, preglednosti i čitanju tako bilježenih podataka, a posebice u analiziranju te otkrivanju veza i utjecaja različitih pokazatelja na razvoj bolesti i stanje pacijenta. Istraživanja i napredak u području računarstva zadnjih nekoliko desetljeća omogućili su razvoj tehnologije pa tako i senzorskih uređaja. Oni mogu automatski mjeriti i prenositi većinu ključnih fizioloških parametara neinvazivnim putem, a programskom podrškom moguće ih je integrirati u veće zdravstvene sustave gdje je omogućena nesmetana pohrana i korištenje podataka u stvarnom vremenu, kao i praćenje profila značajnih parametara i životnih navika u odnosu na zdravstveno stanje pacijenta. Razvojem zdravstveno-informatičkih sustava i programskih rješenja nastoji se olakšati i unaprijediti upravljanje vlastitim zdravljem ili bolešću te potaknuti pacijente na aktivno uključivanje, djelotvornije upravljanje dijabetesom i pojačanu kontrolu nad vlastitim zdravljem. S druge strane, nastoji se povezati zdravstvene

ustanove i liječnike sa pacijentima i na taj način omogućiti kontinuirano udaljeno praćenje stanja pacijenata, personalizirano liječenje i upravljanje dijabetesom te pravovremeno donošenje odluka, a pacijente usmjeriti i osnažiti u preuzimanju veće odgovornosti za vlastitim zdravljem (engl. “*patient empowerment*”).

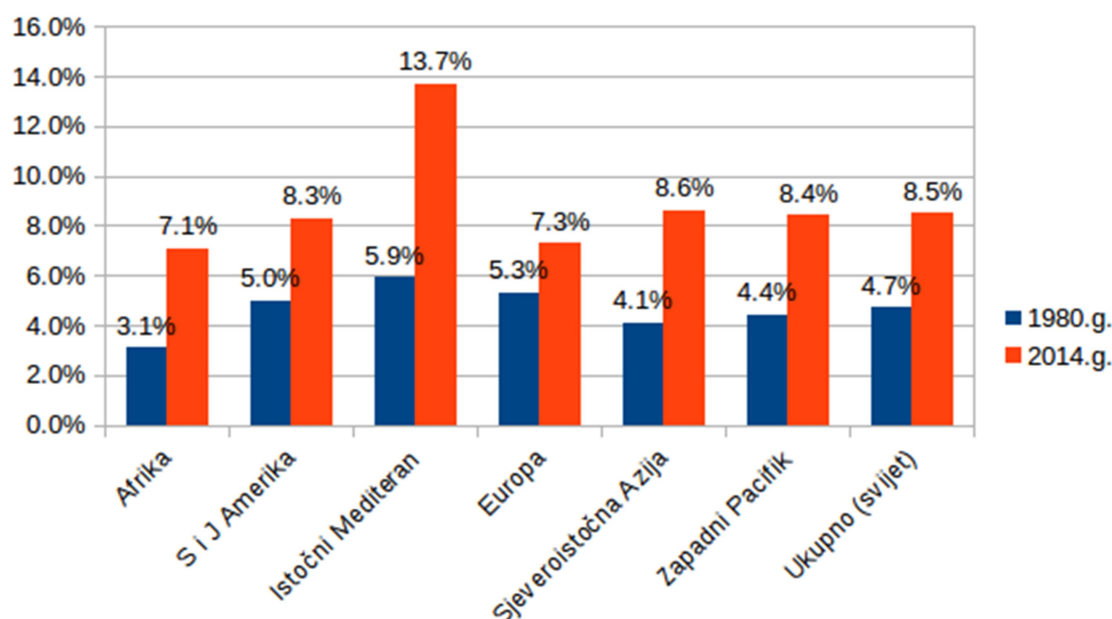
Na početku drugog poglavlja rada opisan je dijabetes i utjecaj ove bolesti na pacijente. U tom poglavlju govori se i o utjecaju dijabetesa na globalnoj razini i u Hrvatskoj. Spominju se dijagnostički i terapijski postupci u kojima se razmatraju faktori rizika koji mogu prouzrokovati razvoj bolesti te metode koje se koriste u dijagnozi i otkrivanju ove bolesti. Nakon toga govori se o prevenciji i liječenju gdje se opisuju ključni postupci u sprječavanju prerastanja predijabetesa u dijabetes te načini liječenja i utjecaj različitih vrsta terapija na bolest i život oboljelog. Zatim se opisuju važni pokazatelji zdravstvenog stanja koji utječu na razinu šećera u krvi i njihovo praćenje što je potrebno u potpunosti razumjeti kako bi bilo koji zdravstveno-informatički sustav bio upotrebljiv i zadovoljavao potrebe korisnika i zdravstvenih djelatnika. Na kraju ovog poglavlja govori se o dosadašnjim istraživanjima i eksperimentima u aspektu primjene ICT u zdravstvu te se opisuju postojeća popularna programska rješenja. U poglavlju “Programsko rješenje za praćenje zdravstvenog stanja” opisuje se razvijeno programsko rješenje koje demonstrira učinkovitu upotrebu i spoj različitih tehnologija u zdravstvu, odnosno upravljanju dijabetesom. Ovdje se opisuju korištene računalne platforme, alati i tehnologije koje omogućuju rad i korištenje cjelokupnog sustava. Opisani su dijelovi sustava i načini na koji su oni povezani u cjelinu. Također, u ovom poglavlju govori se i o razvoju i načinu rada sustava gdje se opisuju značajke poput baze podataka te sinkronizacija s uređajima koji su iskorišteni za automatizirano prikupljanje podataka. Opisuje se razvoj web i mobilne aplikacije kao i programske podrške na strani poslužitelja koja omogućuje rad cjelokupnog sustava, a na kraju je detaljnije objašnjena i sinkronizacija uređaja za mjerenje glukoze u krvi sa strane Android mobilne aplikacije. Posljednje poglavlje predstavlja zaključak ovog rada.

2. DIJABETES I PROBLEMI KOJE DONOSI

Dijabetes ili šećerna bolest je poremećaj metabolizma koji nastaje zbog prevelike količine krvnog šećera (glukoze) u krvi. Glukoza je glavni izvor energije u ljudskom tijelu koja se stvara unosom hranjivih tvari u organizam. Hranjive tvari sastoje se od ugljikohidrata, masti i bjelancevina. Ugljikohidrati predstavljaju najbolji izvor energije, a gotovo sav udio unesenih ugljikohidrata pretvara se u glukozu. Glukoza u krv dolazi iz probavnih organa razgradnjom hrane ili iz jetre, a hormon koji pomaže glukozu da iz krvi uđe u stanice naziva se inzulin. Inzulin se stvara u gušterači, a veća razina šećera u krvi potiče gušteraču na veće stvaranje inzulina. Kada glukoza iz krvi uđe u stanice, one ju koriste za proizvodnju energije koja je nužna za rad organizma te se time ujedno smanjuje razina šećera u krvi. Organizam oboljelog od dijabetesa inzulin ne proizvodi u dovoljnoj količini, uopće ga ne proizvodi ili ga ne koristi ispravno. Iz tih razloga glukoza ostaje u krvi i nema mogućnost ulaska u stanice, a povećana i prekomjerna razina šećera u krvi uzrokuje nagle i prolazne reakcije, ali i ozbiljne dugoročne zdravstvene komplikacije poput srčanih i bubrežnih bolesti, problema s vidom, dentalnih bolesti i oštećenja živaca [1]. Razina šećera u krvi izražava se u mmol/l ili u mg/dl te za neoboljele iznosi između 4.0 i 6.0 mmol/l prije unosa hranjivih tvari, a nakon unosa do 7.8 mmol/l. Oboljeli od dijabetesa trebaju voditi računa da njihove vrijednosti razine šećera prije unosa hranjivih tvari variraju između 4.0 i 7.0 mmol/l, a nakon unosa ne prelaze 9.0 mmol/l [2].

Postoje dva najzastupljenija tipa dijabetesa: dijabetes tip 1 (ovisan o inzulinu) i dijabetes tip 2 (neovisan o inzulinu). Organizmi oboljelih od dijabetesa tipa 1 ne proizvode inzulin, već imuni sustav uništava stanice gušterače zadužene za proizvodnju inzulina. Ovaj tip najčešće je dijagnosticiran kod djece i osoba mlađe dobi, a oboljeli kako bi preživjeli moraju svakodnevno uzimati doze inzulina. Organizmi oboljelih od dijabetesa tipa 2 ne proizvode dovoljno inzulina ili ga ne koriste pravilno. Dijabetes tipa 2 najviše je zapažen kod osoba u srednjim godinama i starijim osobama te predstavlja najčešći oblik dijabetesa. Većina oboljelih od dijabetesa (devet od deset oboljelih) ima dijabetes tipa 2 [1].

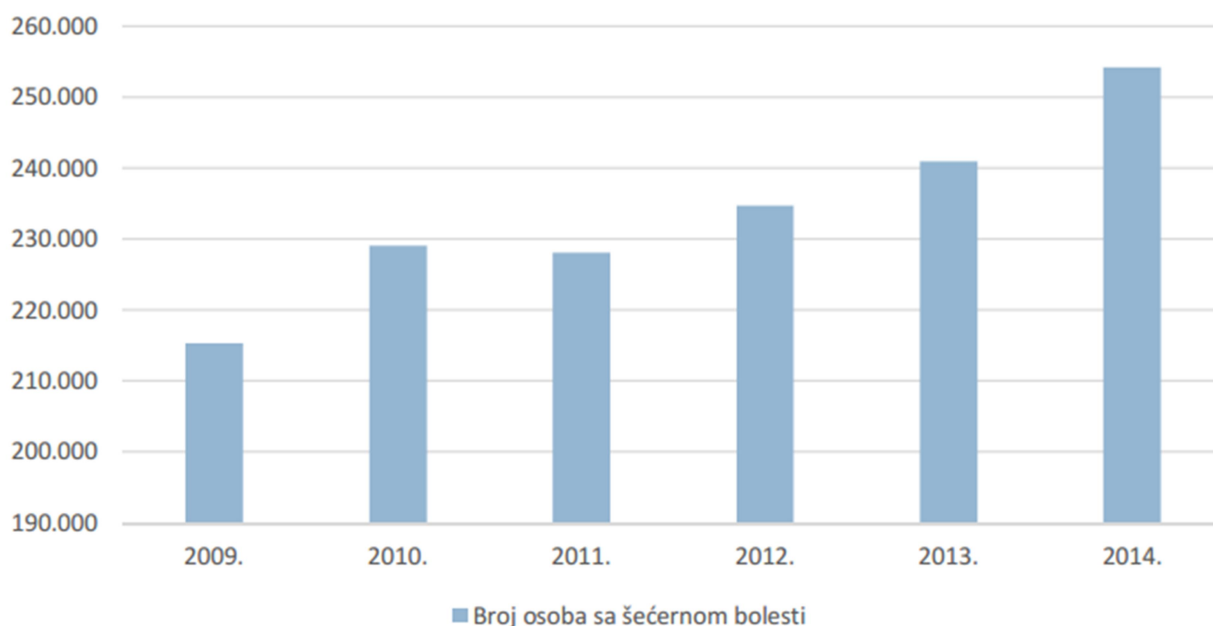
Na globalnoj razini, u 2014. godini prema agenciji World Health Organization (WHO), procijenjeno je da je 422 milijuna ljudi živjelo s dijabetesom, dok je u 1980. godini broj ljudi oboljelih od dijabetesa iznosio 108 milijuna. Rasprostranjenost dijabetesa u populaciji odraslih osoba u svijetu od 1980. godine do danas gotovo se udvostručila, odnosno narasla je s 4.7% do 8.5% kako je prikazano slikom 2.1, izrađenoj prema [3].



Slika 2.1. Graf rasprostranjenosti dijabetesa u 1980. i 2014. godini.

Porast rasprostranjenosti dijabetesa povezan je s povećanjem faktora rizika povezanih s nastankom dijabetesa poput nepravilne ishrane, povećanja tjelesne mase, pretilosti ili fizičke neaktivnosti. Istraživanja su pokazala da je rasprostranjenost fizičke neaktivnosti sve više i više globalno zastupljena. Podaci WHO-a iz 2010. godine pokazali su da 27% žena i 20% muškaraca nisu zadovoljili minimalne potrebe fizičke aktivnosti na tjednoj bazi te su klasificirani kao nedovoljno fizički aktivni. Prikupljeni podaci iz 2014. godine pokazuju da je prekomjerna tjelesna masa i debljina znatno povećana u gotovo svim zemljama. Procijenjeno je da je te godine više od 30% osoba imalo problema s prekomjernom tjelesnom masom, a više od 10% problema s pretilošću. U 2012. godini dijabetes je prouzrokovao smrt 1.5 milijuna ljudi zbog nastalih zdravstvenih komplikacija uzrokovanih previsokom razinom šećera u krvi [3].

U Hrvatskoj je 2014. godine identificirano preko 250 tisuća odraslih osoba sa dijabetesom - 49% muškaraca, te 51% žena. Nasuprot visoke rasprostranjenosti ove bolesti, vidljiv je i znatan rast bolesti tipa 2, a najnoviji podaci pokazuju rasprostranjenost dijabetesa u Hrvatskoj iznosi 6.86% u dobnoj skupini od 20 do 79 godina. Rasprostranjenost šećerne bolesti u razdoblju od 2009. do 2014. godine kretala se u rasponu od 6.69% do 7.90%, odnosno broj oboljelih kretao se između 215 tisuća i 254 tisuće (Slika 2.2.). Najučestaliji oblik liječenja šećerne bolesti u Hrvatskoj u 2014. bio je primjenom hipoglikemika (tablete koje oboljeli od dijabetesa tipa 2 uzimaju kad razina šećera u krvi ne može biti zadovoljavajuće regulirana prehranom, tjelovježbom ili smanjenjem tjelesne mase) te inzulinskih doza.



Slika 2.2. Graf rasprostranjenosti dijabetesa u Hrvatskoj od 2009. do 2014. godine [4].

Prema Hrvatskom zavodu za javno zdravstvo (HZJZ), istraživanje Hrvatskog zavoda za zdravstveno osiguranje (HZZO) pokazalo je da šećerna bolest sa sobom nosi i velika ekonomska opterećenja. Provedena analiza troškova liječenja pokazala je da troškovi po oboljeloj osobi dosežu iznos od gotovo 2000 eura godišnje. To znači da sveukupni izdaci za oboljele predstavljaju 11.49% ukupnih troškova HZZO-a. Također, istraživanje je pokazalo potencijalne znatne uštede i poboljšanu zdravstvenu zaštitu oboljelih intenziviranim pristupom (bolja kontrola, bolja dijagnostika, povećan broj pregleda) i strožom kontrolom bolesti uvođenjem informacijskih tehnologija. Procijenjeno je da bi pojedinac navedenim pristupom mogao živjeti 13% duže - uz sprječavanje mogućih komplikacija i bolju kvalitetu života. HZJZ zaključuje da je važnost informacijskih tehnologija u brizi za oboljele pokazana u analizi financijskih i kliničkih aspekata implementiranja sustava informacijskih tehnologija u kontroli šećerne bolesti [4].

2.1. Problemi s kojima se nose oboljeli

Nepravilno kontroliran ili nekontroliran dijabetes kod oboljelih može voditi do velikog broja kratkoročnih, ali i dugoročnih zdravstvenih komplikacija. Većina komplikacija povezanih s dijabetesom pojavljuju se kao posljedica povećane razine šećera u krvi, a teže komplikacije pojavljuju se nakon dužeg perioda nekontrolirane povećane razine šećera koja u duljem periodu odstupa od optimalnih vrijednosti. Visoka razina šećera u krvi prouzrokuje oštećenje krvnih žila i živaca što dovodi do oštećenja funkcioniranja organizma. Dobrom kontrolom dijabetesa, zdravim životom i aktivnim stilom života dijabetičari mogu smanjiti broj komplikacija ili

spriječiti komplikacije od pojavljivanja te živjeti bez pojave komplikacija i nekoliko desetaka godina.

Kratkoročne komplikacije mogu se pojaviti u bilo kojem trenutku kada razina šećera u krvi značajno padne ili značajno naraste u kraćem periodu vremena. Ova vrsta komplikacija predstavlja neposrednu opasnost i stoga treba brzo i djelotvorno pristupiti liječenju kako bi se izbjegle hitne situacije. Najzastupljenije kratkoročne komplikacije su [5]:

- hipoglikemija
- ketoacidoza (DKA)

Hipoglikemija je stanje koje za razliku od ostalih komplikacija nastupa kada razina šećera u krvi padne ispod 4 mmol/l (72 mg/dl). Iako je dijabetes usko vezan uz znatno povišenu razinu šećera u krvi, nepravilna ishrana te nekontrolirano liječenje i uzimanje lijekova može dovesti do preniske razine šećera u krvi (hipoglikemije) [6]. Osim navedenih uzročnika, alkohol također predstavlja opasnost. Alkoholna pića sadrže velik udio ugljikohidrata što inicijalno povećava razinu šećera u krvi, no jetra koja je zadužena za otpuštanje uskladištenog šećera (glikogena), nakon nekog vremena postaje opterećena razgradnjom alkohola što ju sprječava u otpuštanju tog uskladištenog šećera te razina šećera u krvi počinje naglo opadati. S obzirom da alkohol ima tendenciju da na početku povisi razinu šećera u krvi, a kasnije naglo smanji, oboljeli prevelikom dozom lijeka mogu još više utjecati na kritično smanjenje šećera u krvi što dovodi do hipoglikemije [7]. Glavni simptomi su prekomjerno znojenje, umor, vrtoglavica, a posljedice mogu biti u rasponu od blagih do teških. Blage epizode hipoglikemije obično su tretirane od strane pojedinca te nisu povezane sa značajnim dugoročnim zdravstvenim problemima. Teške epizode zahtijevaju hitno liječenje i brzo postupanje jer mogu dovesti do velikih prijetnji poput kome, gubitka svijesti ili napadaja, a mogu prouzrokovati i smrt. Hipoglikemija se tretira unosom 15-20g brzo razgradivih ugljikohidrata iz glukoznih tableta, slatkiša, slatkih napitaka ili voćnog soka te je nakon 20-ak minuta potrebno provjeriti je li razina šećera u krvi stabilizirana (testiranjem šećera u krvi) [6]. Ketoacidoza je ozbiljna komplikacija koja nastupa kada organizam ostaje bez inzulina, a razina šećera u krvi jako naraste. Uobičajeno nastupa kada šećer u krvi kroz 24 sata ostaje na previsokoj razini (15 mmol/l), to jest znatno većoj od optimalne. Neki od simptoma koji se pojavljuju su povraćanje, dehidracija, neobičan zadah (poput neugodnog mirasa alkohola ili kruške), hiperventilacija, brzi otkucaji srca te zbunjenost i dezorijentacija. U slučaju ketoacidoze potrebna je brza akcija i hitno liječenje jer u suprotnom

ova komplikacija može dovesti do smrti pacijenta. Ketoacidoza se najefikasnije može izbjeći stalnom kontrolom i održavanjem optimalne razine glukoze u krvi [8].

Dugoročne zdravstvene komplikacije u najvećoj mjeri odnose se na:

- probleme s vidom i očne bolesti
- bubrežne probleme
- srčane i krvožilne probleme
- oštećenja živaca i živčanog sustava
- bolesti stopala

Bolesti očiju prouzrokovane dijabetesom mogu voditi prema gubitku vida. Istraživanja pokazuju da regulacija razine šećera u krvi odgađa ili u potpunosti sprječava postepen gubitak vida. Dokazano je da uz razine šećera u krvi veće od preporučene optimalne razine također i visok krvni tlak oštećuje oči pa je stoga bitno da dijabetičari održavaju krvni tlak ispod 130/80 mmHg. Povišena razina šećera u krvi kroz duži period vremena može prouzrokovati i bubrežne bolesti. Bubrezi pomažu organizmu u filtriranju otpada, zvanog urea, i toksina koji ne smiju biti zadržani u organizmu. Nekontroliran dijabetes oštećuje dijelove bubrega zaduženih za filtriranje što vodi k zatajenju bubrega. Osnovni pristup u prevenciji je održavanjem razine šećera u krvi što bliže preporučenoj optimalnoj razini i održavanje krvnog tlaka ispod 130/80 mmHg. Srčani i krvožilni problemi prouzrokovani dijabetesom predstavljaju najčešći uzrok smrti oboljelih od dijabetesa. Uključuju komplikacije poput visokog krvnog tlaka, srčanih udara i nepravilnog rada srca te moždanog udara. Također, uključuju i komplikacije povezane s lošom cirkulacijom krvi u nogama i stopalima. Vrtoglavica, nejasan govor, osjećaj umora ili slabost jedne ruke ili noge mogu ukazivati na pojavu srčanih i krvožilnih problema. Prevencija ovih problema kod dijabetičara moguća je kroz ispravnu ishranu s naglaskom na regulaciju kolesterola i masti, fizičku aktivnost i nepušenje (ili prestanak pušenja). Pušenje predstavlja velik problem kod pacijenata jer i pušenje i povećana razina glukoze oštećuju krvne žile i suzuju ih što u kombinaciji još u većoj mjeri otežava cirkulaciju krvi [9]. Loše upravljan dijabetes je velik uzročnik poremećaja rada živčanog sustava. Dugoročne previsoke razine šećera u krvi, povišene masnoće u krvi i prekomjerna tjelesna masa mogu uzrokovati gubitak osjetila u rukama, nogama i stopalima te živčane komplikacije u probavnom traktu, srcu i spolnim organima. Komplikacije sa stopalima su čest problem kod dijabetičara [10]. Osim činjenice da oštećenje živčanog sustava vodi do gubitka osjećaja u stopalima, kontinuirana previsoka razina šećera u krvi može prouzrokovati lošu cirkulaciju krvi, teže zacjeljivanje rana i povećanu mogućnost infekcija što može voditi pojavljivanju gangrene. Sprječavanje mogućih komplikacija sa stopalima, osim

regulacijom glukoze u krvi, moguće je dnevnom samoprovjerom stopala, pojačanom higijenom te povremenim pregledima od strane specijalista [11].

2.2. Dijagnostički i terapijski postupci

Oboljeli od dijabetesa mogu živjeti dugo i zdravo ukoliko se na pojavu bolesti čim ranije reagira i ukoliko ju se što bolje prati, kontrolira i njome upravlja. Prvi uvjet za dobar život je rana dijagnoza bolesti - što dulje osoba živi s nedijagnosticiranim i nepravilno tretiranim dijabetesom zdravstveni ishodi sve su gori. Dijabetes tipa 1 često dolazi sa simptomima poput pojačane žeđi, gubitka tjelesne mase i obilnog mokrenja koji potiču pacijenta da kontaktira zdravstvene službe. Dijabetes tipa 2 često ne pokazuje alarmantne simptome te ga oboljeli mogu otkriti dosta kasno kroz komplikacije poput naglog i iznenadnog mršavljenja, slabljenja vida, problema sa srcem, povećanom glađu te povećanim umorom. Ovaj tip razvija se vrlo sporo i često postoji vrlo dugačak period u kojem je bolest prisutna ali neotkrivena [3].

2.2.1. Faktori rizika i dijagnoza

Kod osoba kod kojih nisu vidljivi znakovi dijabetesa, debljina, pretilost i nedostatak fizičke aktivnosti mogu voditi razvoju inzulinske rezistencije, odnosno otpornosti organizma na inzulin što se dalje može pretvoriti u dijabetes. Debljina, osobito višak masnoća oko struka, primarni je uzročnik otpornosti na inzulin. Kontinuiranim prekomjernim unosom hrane (energije) pojačano se stvaraju loše masnoće u krvi koje prelaze uobičajene vrijednosti (LDL kolesterol), a smanjuju se korisne masnoće (HDL kolesterol). Drugim riječima, povećanom potrebnom za inzulinom i povećanim masnoćama u krvi, organizam postaje otporniji na inzulin te se šećerna bolest puno lakše može razvijati. Znanstvenici još uvijek u potpunosti ne razumiju zašto se dijabetes tipa 2 kod nekih ljudi razvije, a kod nekih ne. S druge strane, jasno je i sigurno koji faktori rizika povećavaju mogućnost razvoja dijabetesa [12]:

- prekomjerna tjelesna masa ili pretilost - što više masnog tkiva tijelo sadrži, stanice postaju sve otpornije na inzulin
- fizička neaktivnost - fizička aktivnost pomaže u kontroli tjelesne mase, potiče na veće korištenje energije iz glukoze te stanice postaju osjetljivije na inzulin
- obiteljska povijest - rizik od pojave dijabetesa raste ako roditelj ili netko iz uže obitelji ima dijabetes
- rasa - još nije jasno zašto, ali afroamerikanci, latinoamerikanci, amerčki indijanci i azijski amerikanci imaju veće šanse za razvojem dijabetesa

- godine - rizik od dijabetesa raste kako osoba stari, pogotovo nakon 45. godine života, a razlog tome je vjerojatno zbog manje fizičke aktivnosti, smanjenja udjela mišićne mase i povećanja tjelesne mase kroz godine
- predijabetes - stanje u kojem je razina šećera u krvi viša od normalnih granica, ali ne dovoljno visoka da bude klasificirana kao dijabetes (ukoliko se ne liječi na vrijeme, predijabetes često prerasta u dijabetes tipa 2)

Osoba koja je u stanju predijabetesa može spriječiti razvoj bolesti ukoliko je to stanje otkriveno na vrijeme. Agencija American Diabetes Association (ADA) preporuča testiranje na predijabetes (dijabetes) svim odraslim osobama koje su deblje ili pretile i imaju barem jedan ili više od prethodno navedenih faktora rizika dijabetesa uključujući povećani krvni pritisak (140/90 mmHg ili više), razinu HDL kolesterola ispod 35 mg/dl ili kroničan stres. Ukoliko su rezultati testiranja normalni, testiranje bi se trebalo ponoviti svake tri godine. Predijabetes se može dijagnosticirati krvnim testovima poput A1c testa još poznatog i kao test koncentracije hemoglobina A1c ili kao HbA1c test, FPG test (test glukoze u plazmi, engl. *fasting plasma glucose test*) ili OGTT testom (oralni glukoza tolerans test, engl. *oral glucose tolerance test*) [13].

A1c test je najpouzdaniji test kojim se u laboratoriju ispituje koncentracija hemoglobina A1c (proteinska molekula u crvenim krvnim stanicama) koja se mijenja kada se na nju veže glukoza. Ovim ispitivanjem zapravo se mjeri srednja razina glukoze u krvi u zadnja tri mjeseca, a vrlo je precizan jer nedavna promjena ishrane ne utječe na rezultate. Rezultat se izražava kao postotak, a često se prikazuje samo kao brojčana vrijednost koja predstavlja procijenjenu prosječnu razinu glukoze (eAG, engl. *estimated Average Glucose*). Vrijednosti ispod 5.7% određuju osobu koja nema predijabetes i kod koje je razina glukoze normalna. Veće vrijednosti vrijednosti znače višak glukoze što je povezano s dijabetičkim komplikacijama [14]. FPG testom mjeri se razina glukoze kod ljudi koji u zadnjih 8 sati nisu ništa jeli, a najprecizniji je u jutarnjim satima. Razine između 100 mg/dl i 125 mg/dl predstavljaju predijabetes. OGTT je vrsta testa u kojoj se u laboratoriju mjeri razina glukoze nakon što osoba nije jela barem 8 sati, a vrijednost između 140 i 199 mg/dl predstavlja predijabetes. U sljedećoj tablici vidljive su vrijednosti različitih vrsta testova.

Tablica 2.1. Odnos vrijednosti i stanja različitih vrsta testova za dijagnozu.

	A1c [%]	FPG [mg/dl]	OGTT [mg/dl]
Dijabetes	6.5 ili više	126 ili više	200 ili više
Predijabetes	5.7 do 6.4	100 do 125	140 do 199
Normalno stanje	oko 5	99 ili manje	139 ili manje

Rezultati ispitivanja koji predstavljaju predijabetes znači da je otpornost organizma na inzulin napredovala te nije u stanju proizvesti dovoljne količine inzulina što prouzrokuje rast razine šećera u krvi i vodi prema dijabetesu tipa 2. Veće vrijednosti testa predstavljaju i veći rizik od pojavljivanja dijabetesa, a razina opasnosti također ovisi i o prethodno spomenutim faktorima rizika [13].

2.2.2. Prevencija i liječenje

Ključ u prevenciji prerastanja predijabetesa u dijabetes ili u liječenju dijabetesa je u sljedećim postupcima:

- zdrava ishrana i održavanje zdrave tjelesne mase
- povećanje fizičke aktivnosti
- kontrola razine šećera u krvi
- smanjenje unosa alkohola i odbacivanje loših navika (poput pušenja)
- smanjenje stresa
- kontrolirano uzimanje lijekova poput tableta ili doza inzulina

U pogledu zdrave prehrane, oboljelima je bitno usredotočiti se na hranu s niskim udjelom masti i visokim udjelom vlakana, poput voća, povrća i cjelovitih žitarica. Bitno je i smanjiti unos hrane životinjskog podrijetla, a pogotovo unos rafiniranih ugljikohidrata, odnosno hrane koja ima visok glikemijski indeks (poput slatkiša, slatkih pića, bijelog kruha, bijele tjestenine,...). Glikemijski indeks (GI) je mjera koja pokazuje koliko brzo hrana uzrokuje povećanje šećera u krvi [14]. Hrana s visokim GI povećava krvni šećer vrlo brzo zbog teže probave te hrane, dok s druge strane hrana s niskim GI rezultira s umjerenim i mirnijim otpuštanjem glukoze te pomaže u ostvarivanju stabilnije razine šećera u krvi. Hrana s niskim GI pomaže u mršavljenju, dok hrana s visokim GI ima korist u liječenju hipoglikemije.

Tablica 2.2. Različiti iznosi glikemijskog indeksa u vezi s dodatnim pojavama.

	Vrlo nizak GI	Nizak GI	Srednji GI	Visok GI
Vrijednost	0-35	36-55	56-69	70+
Dodatne pojave	mršavljenje	uravnotežena tjelesna masa	debljanje	ubrzano debljanje

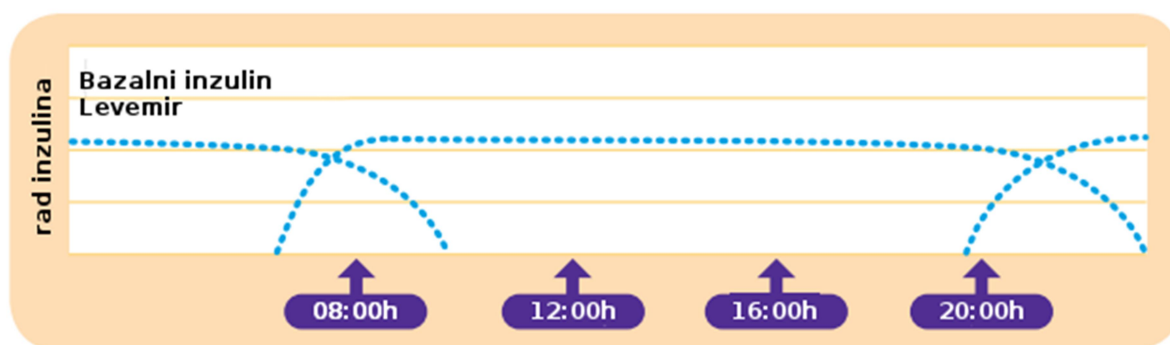
Održavanje tjelesne mase u normalnim granicama često je primarni cilj kod oboljelih od dijabetesa. Kod pacijenata s prekomjernom tjelesnom masom, gubitak težine može pomoći u poboljšanju osjetljivosti na inzulin te olakšati upravljanje dijabetesom. Kako bi se to postiglo, potrebno je pridržavati se niskokalorične prehrane i uvesti veću količinu povrća u prehranu jer sadrži raznolike hranjive tvari, a s obzirom da sadrži i velik udio vlakana, pomaže u zadržavanju osjećaja sitosti i smanjenju kolesterola. Fizička aktivnost može biti vrlo učinkovita u kontroli razine šećera u krvi. Kada mišići rade, koriste glukozu iz krvi, jetre i mišića, a nakon aktivnosti organizam obnavlja uskladištenu glukozu uzimanjem i trošenjem dostupne glukoze iz krvi [16].

Većina pacijenata za upravljanje dijabetesom i liječenje treba uzimati dodatne doze inzulina ili druge lijekove i pripravke protiv dijabetesa. Odabir vrste lijeka ovisi o tipu dijabetesa i drugim faktorima poput zdravstvenog stanja i načina života. Oboljeli od dijabetesa tipa 1 moraju uzimati doze inzulina nekoliko puta tijekom dana jer ga njihov organizam uopće ne proizvodi. Oboljeli od dijabetesa tipa 2 trebaju dodatne doze inzulina u manjoj mjeri te se mogu liječiti i uzimanjem tableta za liječenje dijabetesa. Postoji više vrsta inzulina koje se razlikuju po brzini i trajanju djelovanja (tablica 2.3.). Većina vrsta doseže maksimum koji predstavlja najjači učinak djelovanja. Primjena određene vrste ovisi o preporuci liječnika i situaciji u kojoj se oboljeli nalazi [17].

Tablica 2.3. Različite vrste inzulina i njihove karakteristike.

Vrsta inzulina	Vrijeme početka djelovanja	Maksimum djelovanja	Trajanje djelovanja
Ultrakratkodjelujući	nakon 15 minuta	1 sat	2 do 4 sata
Kratkodjelujući	kroz 30 minuta	2 do 3 sata	3 do 6 sat
Srednjedugodjelujući	nakon 2 do 4 sata	4 do 12 sati	12 do 18 sati
Dugodjelujući	nakon nekoliko sati	nema	24 sata (neki i dulje)

Za odlučivanje i prilagodbu potrebnih doza inzulina potrebno je vrijeme te ovisi o osjetljivosti tijela na inzulin (odnosno kako smanjenje razine šećera ovisi o dozi inzulina), potrebama, načinu života i ostalim okolnostima pojedine osobe. Kod dijabetesa liječnik propisuje procijenjenu količinu i vrstu inzulina koju pacijent treba uzimati tijekom dana. Postoje dvije terapije koje su bitne kod doziranja inzulina: bazalna i bolus terapija. Bazalna predstavlja dnevnu potrebnu dozu inzulina potrebnog za održavanje ciljane razine šećera u krvi, a bolus doza predstavlja količinu inzulina potrebnog za pokrivanje unesenih ugljikohidrata, odnosno hrane tijekom dana. Bazalni inzulin (dugodjelujući) kontrolira razinu šećera tijekom dana i za vrijeme sna te se uzima jednom ili dva puta dnevno, dok bolus inzulin (ultrakratkodjelujući ili brzodjelujući) kontrolira povećanje razine šećera tijekom obroka [18]. Levemir je tip bazalnog inzulina koji traje 12-16 sati te se ubrizguje dva puta dnevno u razmaku od 12 sati, kao što je to vidljivo na slici 2.3. Uz bazalnu terapiju moguće promjene razine šećera u krvi tijekom dana reguliraju se bolus terapijom, odnosno dodatnim dozama [21].



Slika 2.3. Djelovanje i rad bazalnog inzulina (Levemir) [18].

U slučaju veće razine šećera u krvi od ciljane, kod oboljelih jedna jedinica ultrakratkodjelujućeg inzulina smanjuje razinu šećera u krvi za 2 mmol/l, međutim, kod nekih je smanjuje za 1 mmol/l, a kod nekih jedna jedinica smanjuje za 3-5 mmol/l. Djelovanje jedinice

inzulina naziva se faktor osjetljivosti inzulina CF (engl. *correction factor*), a procijeniti se može sljedećom formulom:

$$CF = 100 / TDD \quad (2-1)$$

gdje je TDD ukupna dnevna doza inzulina (engl. *total daily dose*). Primjerice, ukoliko je oboljeloj osobi potrebno 8 jedinica inzulina za vrijeme doručka, 6 za vrijeme ručka, 10 za vrijeme večere te 8 jedinica dugodjelujućeg inzulina ujutro i 18 jedinica u 22 sata, slijedi da je $TDD = 8 + 6 + 10 + 8 + 18 = 50$, a $CF = 100/50 = 2$ mmol/l [19]. TDD može se procijeniti i unaprijed na temelju indeksa tjelesne mase, kao što je to prikazano u tablici 2.4.

Tablica 2.4. Procjena TDD vrijednosti na temelju iznosa BMI

BMI < 18.5 (pothranjenost)	18.5 < BMI < 25 (idealno)	25 < BMI < 30 (prekomjerno)	BMI > 30 (pretilost)
0.3 jedinica/kg/dan	0.4 jedinica/kg/dan	0.5 jedinica/kg/dan	0.6 jedinica/kg/dan

Prema navedenom, BMI 25-godišnjaka visokog 180 cm i mase 75 kg iznositi će 23, što rezultira TTD vrijednošću od 0.4 jedinica/kg/dan, to jest 30 jedinica na dan [20]. Postoje razni proračuni potrebnih doza inzulina za liječenje dijabetesa poput izračuna potrebnih jedinica prije ili nakon jela ili ukupnih dnevnih potrebnih doza, no s obzirom da proračuni nisu uvijek potpuno precizni, potrebno je savjetovanje s liječnikom. Liječnik pomaže u određivanju doza bazalnog dugodjelujućeg inzulina te u određivanju dodatnih (ultra)kratkodjelujućih doza prema različitim pokazateljima zdravstvenog stanja oboljelog od dijabetesa.

Osim liječenja dozama inzulina, česta je primjena određenih tableta poput metformin tableta. One smanjuju razinu glukoze koju stvara jetra, pomažu organizmu da inzulin koristi efikasnije te pomažu u smanjenju tjelesne mase. Kombiniranjem više različitih lijekova protiv dijabetesa razina šećera u krvi može se ustabiliti djelotvornije nego korištenjem samo jednog lijeka.

2.3. Važni pokazatelji zdravstvenog stanja i njihovo praćenje

Kako bi se izbjegle komplikacije koje dijabetes može stvoriti, najvažnije je praćenje i upravljanje razinom šećera u krvi. Kako bi se moglo upravljati razinom šećera u krvi, bitno je kontinuirano pratiti i upravljati ostalim pokazateljima koji utječu na nju, a prema [9, 22-25] to su:

- **ABC - A1C, krvni tlak (engl. *blood pressure*), kolesterol (engl. *cholesterol*)**

Odnosi se na A1C test, krvni tlak, kolesterol i prestanak pušenja. A1C testom provjerava se prosječna razina šećera kroz protekla tri mjeseca, a ciljana vrijednost za većinu oboljelih je manja od 7 posto. Ciljanu vrijednost određuje liječnik svaka tri mjeseca. Krvni tlak za većinu oboljelih treba iznositi manje od 140/90 mmHg. Ciljanu vrijednost određuje liječnik te se svakim posjetom liječniku provjeravaju mjerenja kako bi se utvrdila odstupanja i eventualno postavila nova ciljana vrijednost. Kolesterol je supstanca slična masti koja se nalazi u organizmu, a služi za različite metaboličke procese. HDL-kolesterol ("dobar" kolesterol) nastaje u organizmu te služi za "čišćenje" organizma od suvišnih masti, dok LDL tip kolesterola ("loš" kolesterol) odlaže masti, a povećava se unosom zasićenih masnoća u organizam. Prekomjernim unosom masnoća povećava se i ukupna razina kolesterola, razina HDL-kolesterola postaje premala, LDL-kolesterola previsoka te to dovodi do nakupljanja masti u krvnim žilama što uzrokuje otežanu i usporenu cirkulaciju krvi (začepljenje krvnih žila). Poželjna vrijednost ukupnog kolesterola je manje od 5.2 mmol/l, LDL-kolesterola manje od 3.0 mmol/l, a HDL-kolesterola 1.5 mmol/l ili više (minimalno jedna trećina ukupnog). Ciljanu vrijednost određuje liječnik. Ukoliko oboljeli koristi duhanske proizvode, prestanak može poboljšati razinu kolesterola i krvni tlak. Konzumacija duhanskih proizvoda treba se mjeriti isto kao i svi ostali faktori.

- **razina šećera prije obroka**

Za većinu oboljelih treba iznositi od 4 do 7 mmol/l. Ciljanu vrijednost određuje liječnik.

- **razina šećera 2 sata poslije obroka**

Za većinu oboljelih treba iznositi manje do 9 mmol/l. Ciljanu vrijednost određuje liječnik.

- **prehrana i alkohol**

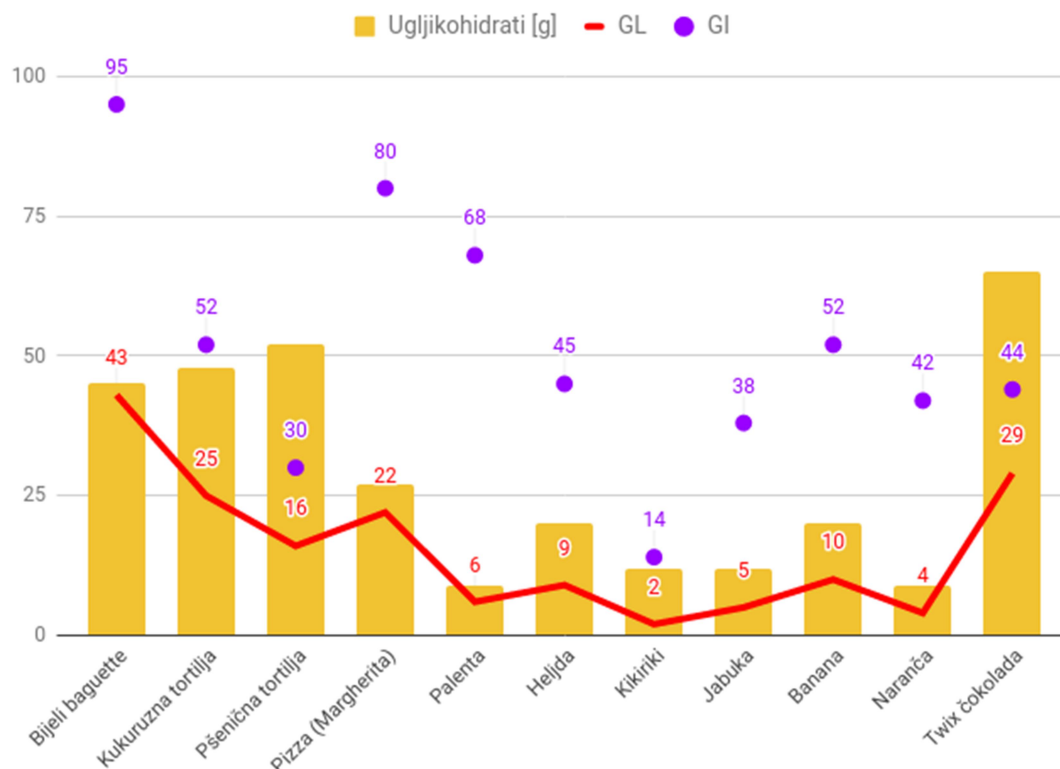
Praćenje dnevnog unosa kalorija te praćenje i odabir ispravne prehrane pomaže u upravljanju razinom šećera u krvi te tjelesnom masom. S obzirom da sastav hrane u velikoj mjeri utječe na dijabetes, praćenje prehrane trebalo bi se odvijati zajedno sa praćenjem doza lijekova i razine šećera u krvi. Bitni faktori su naziv obroka (namirnica), količina te nutritivne vrijednosti (kalorijska vrijednost, udio ugljikohidrata, masti, proteina i vlakana). Za svaki prehrambeni proizvod znaju se ili su naznačene sve spomenute vrijednosti. Količina dnevne potrebne energije izražava se u kalorijama, a ovisi o starosti, spolu, tjelesnoj masi, visini i prosječnoj dnevnoj aktivnosti. Minimalna količina dnevne potrebne energije koja je potrebna organizmu naziva se bazalni metabolizam (BMR, engl. *basal metabolic rate*), a računa se prema Mifflin - St Jeor formuli:

$$BMR = 10 * masa[kg] + 6.25 * visina[cm] - 5 * starost[god.] + 5 \quad (2-2)$$

Tako za prosječnu ženu BMR iznosi 1500 kalorija, dok za muškarca 1700 kalorija. Dnevni unos ugljikohidrata, s naglaskom na što veći udio složenih ugljikohidrata koji imaju visok udio vlakana, trebao bi iznositi oko 45% dnevnog unosa kalorija, a unos prehrambenih proizvoda s visokim glikemijskim indeksom trebalo bi u što većoj mjeri smanjiti. Kako bi se utjecaj pojedine hrane na razinu šećera u krvi mogao točnije odrediti, nije dovoljno znati samo GI, već je potrebno znati odnos između GI, tj. brzine kojom glukoza ulazi u krv i količine glukoze koja ulazi u krv. Količina glukoze koja će ući u krv najviše ovisi o ugljikohidratima, a veličina koja opisuje spomenuti odnos naziva se glikemijsko opterećenje (GL, engl. *glycemic load*):

$$GL = (ugljikohidrati[g] * GI) / 100 \quad (2-3)$$

GL iznosa 10 ili manje smatra se niskom, a iznosa 20 ili više smatra se vrlo visokim. Primjerice, lubenica ima visok GI (80), međutim porcija lubenice ima vrlo malo ugljikohidrata (6 grama) što rezultira niskim GL (5) [15]. Grafikon koji prikazuje usporedbu količine ugljikohidrata te GI i GL vrijednosti vidljiv je na slici 2.4 izrađenoj prema [25].



Slika 2.4. Graf usporedbe količine ugljikohidrata sa GI i GL vrijednostima.

Prosječna oboljela odrasla osoba svakodnevno bi trebala unositi barem 30 grama vlakana. Masti sadrže vrlo visok stupanj energije te bi dnevni unos masti trebao iznositi između 25% do 35% dnevnog unosa kalorija. Zdrave masnoće mogu povećati razinu "dobrog" kolesterola, a nalaze se u namirnicama poput orašastih plodova i maslinovog ulja. Proteinima treba dulje vremena za razgradnju, kontroliranim unosom mogu pomoći u stabilizaciji razine šećera u krvi, a njihov dnevni unos trebao bi iznositi oko 12% do 20% dnevnog unosa kalorija. S obzirom da su ove vrijednosti procijenjene, na temelju zdravstvenog stanja i ostalih pokazatelja oboljelog, liječnik bi trebao pomoći u upravljanju, kontroli i postavljanju pravila prehrane.

- **fizička aktivnost**

Aktivan život pomaže u cirkulaciji krvi te u kontroli razine šećera i tjelesne mase. Fizička aktivnost može predstavljati tjelovježbu, ali i bilo koji drugi oblik fizičke aktivnosti poput šetnje, plesa, bicikliranja, plivanja, trčanja i slično. U odabiru vrste i intenziteta fizičke aktivnosti može pomoći liječnik, a važno je svakodnevno barem 20 minuta odvojiti na šetnju ili neki drugi oblik aktivnosti (ovisno o stanju oboljele osobe). U istraživanju iz 2012. godine zaključeno je da je povećan broj dnevnih napravljenih

koraka povezan sa smanjenjem A1C vrijednosti. Samo 2600 dnevnih dodatnih koraka smanjuje A1C vrijednost za 0.2%, a veći broj dodatnih dnevnih koraka može još značajnije utjecati na stabilnost razine šećera u krvi. U praćenju fizičke aktivnosti bitno je voditi računa o vrsti aktivnosti, trajanju, brzini otkucaja srca ili intenzitetu, potrošenoj energiji, te datumu i vremenu obavljanja aktivnosti.

- **lijekovi**

Količina i vrijeme uzimanja doza lijekova protiv dijabetesa te korištenje ostalih lijekova utječe na dijabetes. Kao što je već prije spomenuto, oboljeli koji se liječe dozama inzulina, jednom ili dva puta dnevno uzimaju bazalnu dozu inzulina, a prije obroka dozu ultrakratkodjelujućeg inzulina. Za praćenje je potrebno obratiti pažnju na vrijeme uzimanja doze pojedine vrste inzulina. Svaki dan potrebno je pratiti bazalne doze inzulina, a uz pojedini obrok i ostale doze inzulina. Vrijeme (trenutak) uzimanja doze inzulina ima velik utjecaj na samo djelovanje te je idealan trenutak 20 minuta prije obroka. Osim toga, oboljeli mogu uzimati i tablete protiv dijabetesa (poput metformina) jednom do tri puta dnevno u količini od 500 mg do 2000 mg (ovisno o potrebama i uputama liječnika) gdje je potrebno pratiti naziv i vrstu lijeka te datum i vrijeme uzimanja lijeka. Miješanje lijekova protiv dijabetesa s lijekovima protiv drugih bolesti može poremetiti rad lijekova protiv dijabetesa i loše utjecati na razinu šećera u krvi, stoga je potrebno pratiti i unos ostalih lijekova.

- **biološki pokazatelji**

Odnose se na stres, bolesti te količinu sna. Stres i bolesti potiču stvaranje glukoze i povećanje razine šećera u krvi. Tjelovježba ili fizička aktivnost te provođenje vremena na zraku može pomoći u smanjenju i kontroli stresa. Kod nekih dijabetičara organizam može postati osjetljiviji na inzulin pojavom neke bolesti što može prouzročiti prenisoku razinu šećera u krvi (kontinuirano praćenje razine šećera u krvi za vrijeme bolesti je nužno). Za vrijeme bolovanja potrebno je povećati unos tekućine (barem jedna čaša vode svaki sat), pridržavati se ispravne prehrane te konzultirati se s liječnikom u vezi lijekova protiv prehlade ili gripe (zbog mogućeg međudjelovanja s lijekovima protiv dijabetesa), a potrebno je voditi računa o iznosu tjelesne mase (svakodnevno), unosu dnevne tekućine (svaku večer), tjelesnoj temperaturi (ujutro i navečer) te ostalim pokazateljima poput doza inzulina (prije svakog obroka) te razine šećera u krvi (prije i poslije obroka). Kvaliteta sna utječe na dijabetes - nedostatak sna ili nekvalitetan san može uzrokovati

težu kontrolu dijabetesa, veću otpornost na inzulin te povećanje tjelesne mase. Umor tijekom dana, a posebice ujutro rezultira povećanjem razine šećera u krvi, a povećanje kvalitete sna moguće je kroz održavanje optimalne razine šećera u krvi, održavanje prostorije ugodnom (temperatura oko 18°C i dobra prozračenost), mračnom i mirnom (bez buke ili vanjskih utjecaja) te održavanja redovitog spavanja (količina i vrijeme). Kod praćenja potrebno je uzeti u obzir vrijeme odlaska u krevet, vrijeme buđenja, procjenu kvalitete sna, broj noćnih buđenja, količinu konzumiranog kofeina ili alkohola prije/poslije spavanja.

- **tjelesna masa i debljina**

Posjetom liječniku trebala bi se provjeriti i postaviti ciljana vrijednost tjelesne mase. Održavanje tjelesne mase u preporučenim granicama pomaže u održavanju i kontroli razine šećera u krvi. Osim same tjelesne mase u kilogramima, prate se i indeks tjelesne mase (BMI) te opseg struka kako bi se dobio bolji pogled na građu tijela oboljelog i kako bi se preispitala debljina ili pretilost što znatno utječe na dijabetes. BMI je okvirni pokazatelj debljine i pretilosti te omogućuje lakše svrstavanje osobe u kategoriju kojoj pripada na temelju vlastite mase i visine. Formula prema kojoj se računa je sljedeća:

$$BMI = m / h^2 \quad (2-4)$$

Vrijednost do 18.5 predstavlja pothranjenost, od 18.5 do 25 idealnu tjelesnu masu, od 25 do 30 prekomjernu tjelesnu masu, a od 30 nadalje pretilost. Opsegom struka i bokova u nekim situacijama može se točnije procijeniti zdravlje u pogledu građe tijela jer se više promatra udio masnih naslaga tijela. Mjera struka može se uspoređivati s visinom (WHtR, engl. *waist-to-height ratio*). Opis WHtR-a u pogledu građe tijela na temelju odnosa struk:visina dana je u tablici 2.5.

Tablica 2.5. Veza između veličine WHtR i odnosa struk-visina.

Muškarci	Žene	Opis
≤ 0.34	≤ 0.34	mršavost, zdrav WHtR
0.43 do 0.52	0.42 do 0.48	zdrav WHtR
0.53 do 0.62	0.49 do 0.57	povećan udio masti, povećan zdravstveni rizik
≥ 0.63	≥ 0.58	jako povećan udio masti, velik rizik od zdravstvenih komplikacija

- **oči, stopala, zubi**

Znakovi komplikacija dijabetesa mogu se pojaviti na navedenim dijelovima tijela te su potrebni regularni specijalistički pregledi liječnika: oči jednom godišnje, stopala svakih tri ili šest mjeseci te zubi svakih šest mjeseci. U praćenju je potrebno navesti datum i rezultate pregleda.

Praćenje svih prethodno spomenutih pokazatelja može se jednostavnije prikazati sljedećom tablicom (tablica 2.6).

Tablica 2.6. Važni pokazatelji zdravstvenog stanja i praćenje.

Pokazatelj ili faktor	Promatrana vrijednost		Prosječna poželjna vrijednost*	Pregled i provjera ciljeva kod liječnika
ABC	A1C test		< 7 %	svaka 3 mjeseca
	krvni tlak		< 140/90 mmHg	svakih 3 ili 6 mjeseci
	kolesterol	ukupan	< 5.2 mmol/l	jednom godišnje
		HDL	> 1.5 mmol/l	
		LDL	< 3.0 mmol/l	
	pušenje		prestanak/nepušenje	barem jednom godišnje
razina šećera	prije obroka		4 - 7 mmol/l	uz A1C
	2h poslije obroka		< 9 mmol/l	uz A1C
prehrana	naziv obroka/namirnice/jela			barem jednom godišnje (savjetovanje o prehrani češće u slučaju prekomjerne tjelesne mase ili pretilosti)
	količina			
	energija (kalorije)		dnevno minimalno da pokrije BMR	
	ugljikohidrati		225g (45%) dnevno	
	vlakna		>= 30g dnevno	
	masti		60g (25% - 35%) dnevno	
	proteini		80g (12% - 20%) dnevno	
fizička aktivnost	vrsta/naziv aktivnosti		šetnja/trčanje/plivanje/...	barem jednom godišnje (savjetovanje o fizičkoj aktivnosti češće u slučaju prekomjerne tjelesne mase ili pretilosti)
	trajanje		najmanje 20-30 min dnevno (barem 2600 dodatnih koraka šetnjom dnevno)	
	otkucaji srca / intenzitet			
	potrošena energija (kalorije)			
	datum i vrijeme		jednom ili više puta dnevno, barem nekoliko dana u tjednu	
lijekovi	inzulin [jedinica]	bazalna doza	ovisno o osjetljivosti na inzulin i stanju	zajedno s A1C testom ili
		bolus doza		

Pokazatelj ili faktor	Promatrana vrijednost		Prosječna poželjna vrijednost*	Pregled i provjera ciljeva kod liječnika
		vrijeme doze	bazalna ujutro i navečer (ovisno o tipu), bolus 20 min prije obroka	provjerom krvnog tlaka
	tablete protiv dijabetesa (npr. metformin)	količina doze	500 mg - 2000 mg	
		vrijeme	ovisno o potrebi, 1-3 puta dnevno	
	ostali lijekovi	vrsta/naziv	ovisno o međudjelovanju s lijekovima protiv dijabetesa, bolesti i uputama liječnika	
		količina		
		vrijeme		
biološki pokazatelji	stres		što manje moguće	barem jednom godišnje
	bolesti	tjelesna masa (provjera)	svakodnevno, poželjno održati stalnu tjelesnu masu	
		dnevna tekućina	barem 1 čaša vode na sat	
		tjelesna temperatura	ujutro i navečer, poželjno < 38°C	
	san	vrijeme početka i buđenja	regularan odlazak u krevet i buđenje (raspon spavanja 7-9 sati)	
		procjena kvalitete i okoline (temperatura prostorije, vanjski utjecaji)	što bolja kvaliteta sna, temperatura oko 18°C, dobra prozračenost, bez buke izvana	
		br. noćnih buđenja	što manje (0)	
		količina kofeina i alkohola prije/poslije	što manje (0)	
tjelesna masa i debljina	tjelesna masa		takva da BMI bude između 18.5 i 25	svakih 3 ili 6 mjeseci
	BMI (masa/visina^2)		18.5 do 25	
	odnos struk-visina (WHtR)		M: 0.43 do 0.52 Ž: 0.42 do 0.48	
oči, stopala, zubi	infekcije, komplikacije, zdravstveno stanje			jednom godišnje oči, svakih 3 do 6 mjeseci stopala, svakih 6 mjeseci zubi

Za praćenje i upravljanje dijabetesom nužno je da liječnici postave ciljeve i optimalne vrijednosti zdravstvenih pokazatelja, kako bi oboljeli znali kojim vrijednostima trebaju težiti i što trebaju postići kako bi smanjili utjecaj dijabetesa na njihov život. Najvažniji su: razina šećera u krvi prije obroka, razina šećera u krvi dva sata nakon obroka, A1C vrijednost, krvni tlak, razina ukupnog, LDL i HDL kolesterola, tjelesna masa, dnevni unos kalorija te dnevna fizička aktivnost.

2.4. Provedena istraživanja i postojeća programska rješenja

Postoje brojne aplikacije i računalni alati namijenjeni praćenju zdravstvenog stanja oboljelog od dijabetesa koji rade s ručnim unosom podataka ili na temelju sinkronizacije s ostalim uređajima (poput pametnih narukvica) koji pomažu u mjerenju zdravstvenih i fizioloških parametara. Programska rješenja informacijske i komunikacijske tehnologije (ICT) pomažu oboljelima u upravljanju njihovim zdravljem, posebno dijabetesom, te ne prikazuju samo stanje parametara, već pružaju personalizirane preporuke temeljene na ulaznim podacima. Uglavnom potiču korisnike na prilagodbu njihovog ponašanja i odnosa prema bolesti. Istraživanja pokazuju da primjena informacijske i komunikacijske tehnologije u upravljanju i praćenju dijabetesa omogućuje znatna smanjenja razine šećera u krvi, olakšanu kontrolu bolesti te bržu reakciju na potencijalne komplikacije, poput istraživanja razrađenog u znanstvenom radu „mHealth for Development: The Opportunity of Mobile Technology for Healthcare in the Developing World“. U svijetu je provedeno već više pilot projekata u vezi primjene ICT-a na dijabetes, međutim trenutno su u upotrebi uglavnom samo rješenja koja oboljelima omogućuju informativno praćenje njihovog zdravstvenog stanja te nisu integrirana u veće zdravstveno-informatičke sustave [26].

Karakteristike koje se pokazuju kao nužne u ICT programskim rješenjima u kontekstu dijabetesa su: m-zdravlje (engl. *mHealth*), odnosno korištenje mobilne tehnologije i prijenosnih uređaja (poput pametnih telefona ili tableta) za olakšanu interakciju korisnika (pacijenata) i razmjenu podataka sa zdravstveno-informatičkim sustavom, senzorski uređaji potrebni za automatsko prikupljanje podataka pacijenta te korištenje metoda kontrole, samokontrole, informiranja, praćenja, modeliranja i analize skupova podataka osobe s dijabetesom. Unaprjeđenjem tehnologije i istraživanjem sve se više približava potrebama korisnika (u području zdravstva i pacijenata) u upravljanju kroničnih bolesti poput dijabetesa, a projekti i programska rješenja koja su se istaknula u posljednjih nekoliko godina su:

- Few Touch Application (FTA) - projekt napravljen u svrhu istraživanja pristupa liječenju i praćenju dijabetesa pomoću mobilnih i web tehnologija
- CroDiab - hrvatski nacionalni registar osoba sa šećernom bolešću razvijen u obliku računalnog sustava koji se sastoji od CroDiabNET aplikacije za integraciju elektroničkog zapisa o bolesniku i prikupljanje podataka te CroDiabWEB aplikacije koja služi za olakšavanje pristupa vlastitim informacijama pacijentima putem Interneta
- MyNetDiary - platforma koja oboljelima pruža mogućnost informativnog praćenja i kontrole dijabetesa kroz različite digitalne dnevnike te se sastoji od web aplikacije

(usluge) koja se koristi iz web preglednika i mobilne aplikacije prilagođene za rad u Android i iOS operacijskom sustavu

- Glooko - platforma koja nastoji povezati pacijente i zdravstvene djelatnike, razvijena je za kontrolu i upravljanjem dijabetesom, sastoji se od Android i iOS mobilne aplikacije namijenjene pacijentima i web bazirane aplikacije namijenjene zdravstvenim djelatnicima koji su zaduženi za praćenje zdravstvenih pokazatelja pacijenata
- Glucosio - besplatan projekt otvorenog koda (engl. *open source*) koji se sastoji od Android i iOS mobilne aplikacije za upravljanje dijabetesom s posebnim naglaskom na istraživanje, a jedinstven je po mogućnosti anonimnog dijeljenja podataka između oboljelih i znanstvenika u svrhu olakšavanja i napretka istraživanja

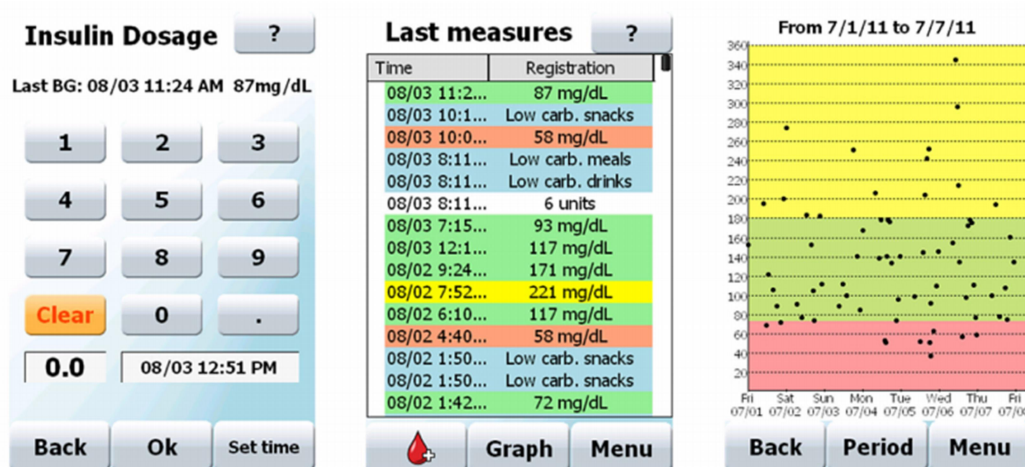
2.4.1. Few Touch Application

Few Touch Application (FTA) je prema Journal of Diabetes Science and Technology časopisu istraživačka platforma razvijena kao alat za pomoć u samokontroli i upravljanju dijabetesom, a kategorizirana je kao osobna zdravstvena aplikacija (PHA, engl. *personal health application*). FTA projektom provedena su istraživanja u kojima su se pratili zahtjevi, potrebe, ponašanja i iskustva krajnjih korisnika (pacijenata) koji su koristili FTA mobilnu aplikaciju. Cilj istraživanja i samog projekta bio je opisati mogućnosti razvoja i korištenje funkcionalnosti mobilnih uređaja za implementaciju m-zdravlje sustava u svrhu liječenja i upravljanja dijabetesom na temelju ispitivanja različitih skupova značajki. Zahtjevi programskog rješenja temeljeni su na sljedećim značajkama: (1) automatski prijenos očitavanja razine glukoze u krvi, (2) razmjena bitnih informacija pomoću SMS (engl. *short message service*) usluge, (3) digitalni dnevnik za praćenje dijabetesa tipa 2, (4) integracija dnevnika s zdravstvenim ustanovama, (5) digitalni dnevnik za praćenje dijabetesa tipa 1, (6) dnevnik fotografija hrane, (7) praćenje fizičke aktivnosti, (8) nutritivne informacije, (9) kontekstna osjetljivost alata i značajki (prilagodba mobilne aplikacije na trenutno okruženje korisnika) i (10) modeliranje razine glukoze u krvi. Bluetooth tehnologija korištena je za automatski prijenos podataka očitavanja razine glukoze u krvi s glukometra (OneTouch Ultra) na mobilni uređaj. SMS usluga korištena je za obavješćivanje roditelja o očitavanjima stanja njihovog djeteta. Svaki put kada bi mobilni uređaj djeteta bio povezan s glukometrom unutar dometa Bluetooth tehnologije (10 m) i kada bi se sinkronizirali podaci očitavanja razine glukoze u krvi, roditeljima bi se poslala SMS poruka sa sadržajem očitavanja stanja njihovog djeteta (Slika 2.5.).



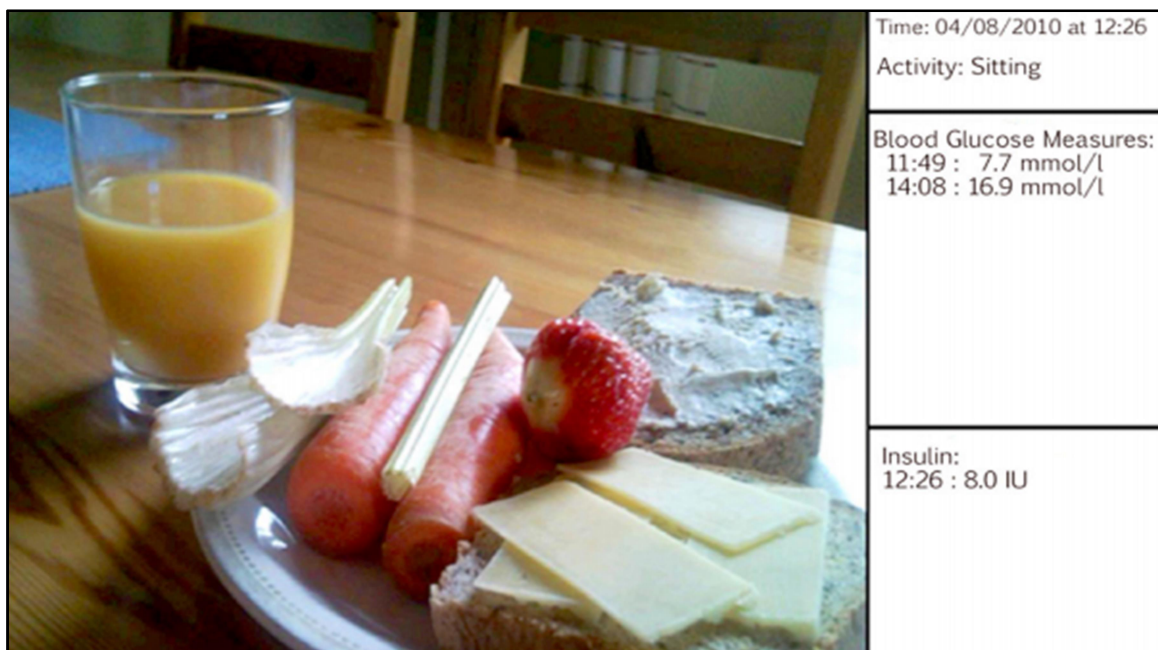
Slika 2.5. Povezanost različitih tehnologija u svrhu praćenja dijabetesa [27].

Dnevnik za praćenje dijabetesa razvijen je tako da sadrži praćenje razine glukoze u krvi (Slika 2.6.), brojač koraka, praćenje unosa hrane i prepoznavanje navika u prehrani, edukacijski sustav s korisnim savjetima, praćenje doza inulina, praćenje napretka u njihovim ciljevima (ciljanim vrijednostima) te grafički prikaz podataka.



Slika 2.6. Digitalni dnevnik za praćenje dijabetesa (razina glukoze u krvi) [27].

Dnevnik fotografija hrane implementiran je kako bi se korisnicima pružio detaljniji i jasniji pogled na vezu između zdravstvenih pokazatelja (poput razine glukoze u krvi) i unosa hrane. Ova značajka uz svaku snimljenu fotografiju hrane automatski uključuje i posljednja relevantna očitavanja razine glukoze u krvi, uzete lijekove (poput doza inzulina) i fizičku aktivnost (Slika 2.7.).



Slika 2.7. Prikaz zapisa iz digitalnog dnevnika fotografija hrane zajedno s ostalim relevantnim podacima [27].

Praćenje fizičke aktivnosti omogućeno je na sličan način kao u nekim prijašnjim značajkama, Bluetooth tehnologijom, gdje se primarno prati broj dnevnih koraka pacijenta. Sinkronizacija podataka odvijala se automatizirano svaku večer, a korisniku je bilo omogućeno da samostalno sinkronizira podatke senzora koraka (uređaja koji mjeri aktivnost) sa mobilnim uređajem, odnosno aplikacijom. Pokazalo se da se korisnicima svidjela jednostavnost ove funkcionalnosti te dodatna motiviranost na kretanje, a neki su izazili želju za funkcionalnošću kojom bi se mjerile i ostale fizičke aktivnosti. Nutritivne informacije pokazale su se vrlo atraktivnim jer su korisnicima omogućile lakši odabir hrane, a pokazalo se i da bi vrlo korisna značajka bila predlaganje namirnica korisnicima na temelju nutritivnih vrijednosti u odnosu na njihovo zdravstveno stanje. Kroz eksperiment je utvrđeno da bi modeliranjem podataka (razine glukoze u krvi, doze inzulina, fizička aktivnost, prehrana,...) uz metode prepoznavanja uzoraka bilo moguće pronaći relevantne uzorke koji bi koristili u pouzdanom predviđanju porasta ili smanjenja razine glukoze u krvi. Osim toga, modeliranjem podataka korisnicima je omogućeno praćenje ovisnosti različitih parametara uz detekciju najboljih i najlošijih ishoda.

Eksperimentiranjem ovim projektom utvrđeno je da je iskustvo korisnika pokazivalo vrlo pozitivne reakcije na sljedeće elemente arhitekturnog dizajna:

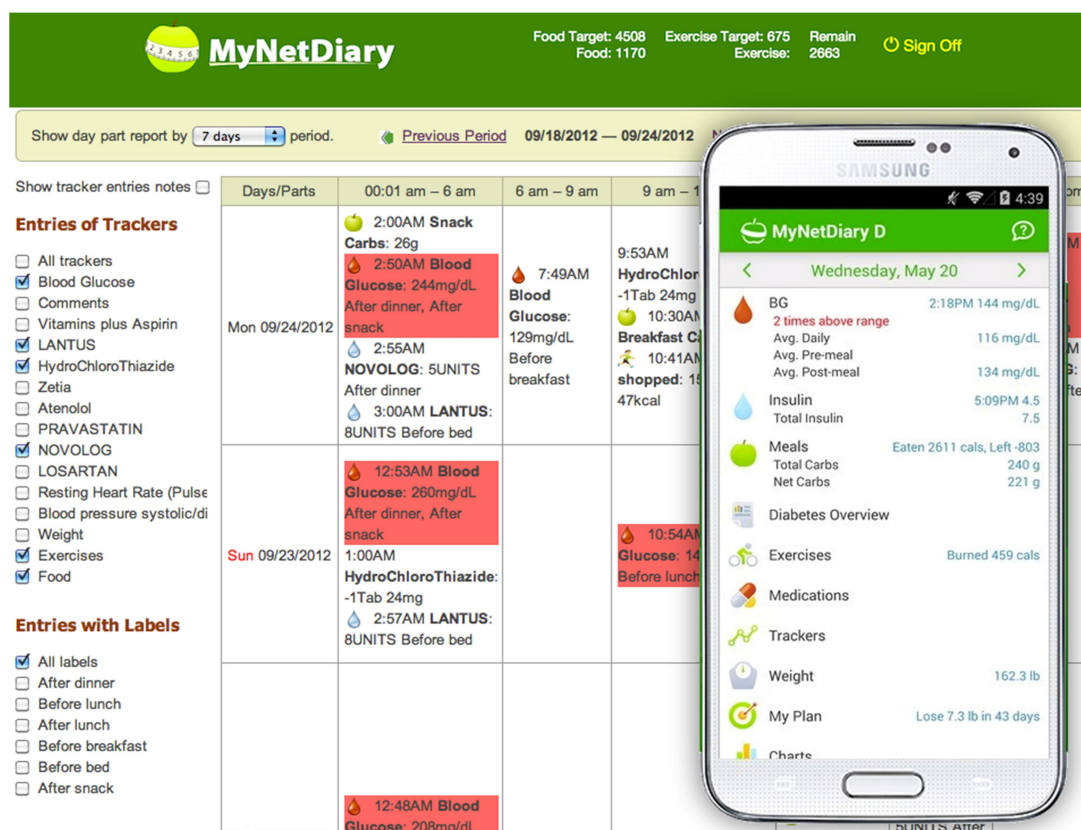
- automatizirano prikupljanje i prijenos podataka kad god je to moguće
- motivacijsko i vizualizirano korisničko sučelje

- pružanje značajne zdravstvene koristi na što jednostavniji način
- dinamičnost: osobna upotreba i upotreba zajedno sa zdravstvenim djelatnicima (integracija u veći zdravstveno-informatički sustav) i slično

Istraživanjem je zaključeno da aplikacije (m-zdravlje) i zdravstveni sustavi ohrabruju i potiču korisnike (pacijente) na veću uključenost i veću motiviranost u upravljanju vlastitim zdravljem te osnažuju ulogu osoba sa dijabetesom [27].

2.4.3. Ostala rješenja u upotrebi

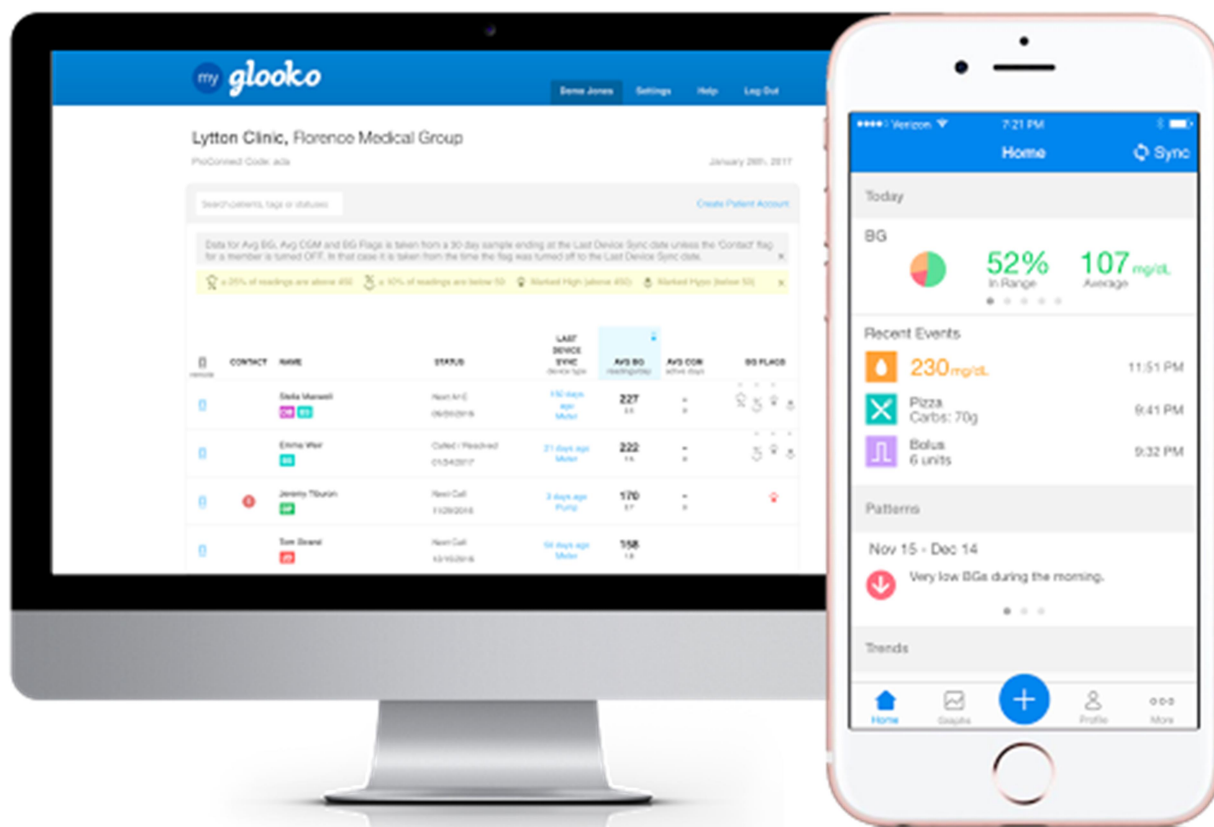
MyNetDiary je platforma koja pruža aplikacije dostupne na Android i iOS platformama, a također pruža i web uslugu kojoj se pristupa preko web preglednika. Funkcionalnosti cjelokupne platforme koncentrirane su na vođenje digitalnog dnevnika prehrane, osobno upravljanje dijabetesom i održavanje ili smanjenje tjelesne mase, odnosno održavanja zdravog načina života. Neke od značajki su zapisivanje i praćenje razina glukoze u krvi, A1C rezultati te praćenje prehrane, fizičke aktivnosti, krvnog tlaka i lijekova (Slika 2.8.). Omogućuje postavljanje alarma (podsjetnika) za provjeru razine glukoze, a nadogradnjom na potpunu verziju nudi i integraciju sa sportskim senzorima (poput pametnih narukvica), planiranje prehrane, detaljniju vizualizaciju podataka i izvještaje. Istaknute prednosti su mogućnost povezivanja s uređajima za mjerenje razine glukoze u krvi, velika baza podataka namirnica i nutritivnih vrijednosti te mogućnost postavljanja pitanja specijalistima na internetskom forumu. Usluge ove platforme naplaćuju se jednokratno, a potpuna verzija naplaćuje se na mjesečnoj ili godišnjoj bazi [28].



Slika 2.8. MyNetDiary web i mobilno programsko rješenje [28].

Glooko je platforma koja povezuje različite tehnologije i nudi softver kao uslugu (engl. *Software as a Service*, SaaS). Sastoji se od Android i iOS mobilne aplikacije te web zasnovane usluge. Ova platforma za upravljanje i kontrolu dijabetesa nastoji povezati pacijente i zdravstvene djelatnike kako bi udaljeni nadzor i praćenje zdravstvenog stanja pacijenata poboljšao rezultate te smanjio troškove liječenja i hitnih slučajeva (Slika 2.9.). Zahvaljujući vlastitoj inovaciji MeterSync Blue uređaja koji povezuje glukometar s mobilnom aplikacijom, Glooko platforma kompatibilna je s velikim brojem glukometara poput Accu-chek, Bayer, Glucocard i OneTouch modela. Korisnička mobilna aplikacija osim sinkronizacije s glukometrom nudi praćenje prehrane i fizičke aktivnosti. Omogućen je prikaz prilagođenih grafova i prosječnih vrijednosti kroz željeni period (dan, tjedan, mjesec, 60 dana, itd.) i generiranje tjednih izvještaja praćenja i mjerenja. Velika prednost Glooko platforme prema drugima je povezanost pacijenta sa zdravstvenim djelatnicima. Kroz svoj sustav nude povezivanje sa zdravstvenim ustanovama koje tada mogu pratiti zdravstvena stanja pacijenata od stanja razine glukoze u krvi i trendova do unosa hrane, lijekova, itd. Također, unutar sučelja, zdravstvenim djelatnicima osim praćenja stanja omogućeno je i postavljanje prilagođenih "zastavica rizika" koje služe za obavješćavanje o kritičnom ili rizičnom stanju pojedinog pacijenta. Iako se platforma još uvijek razvija i ima nedostataka, funkcionalnosti i arhitekturni

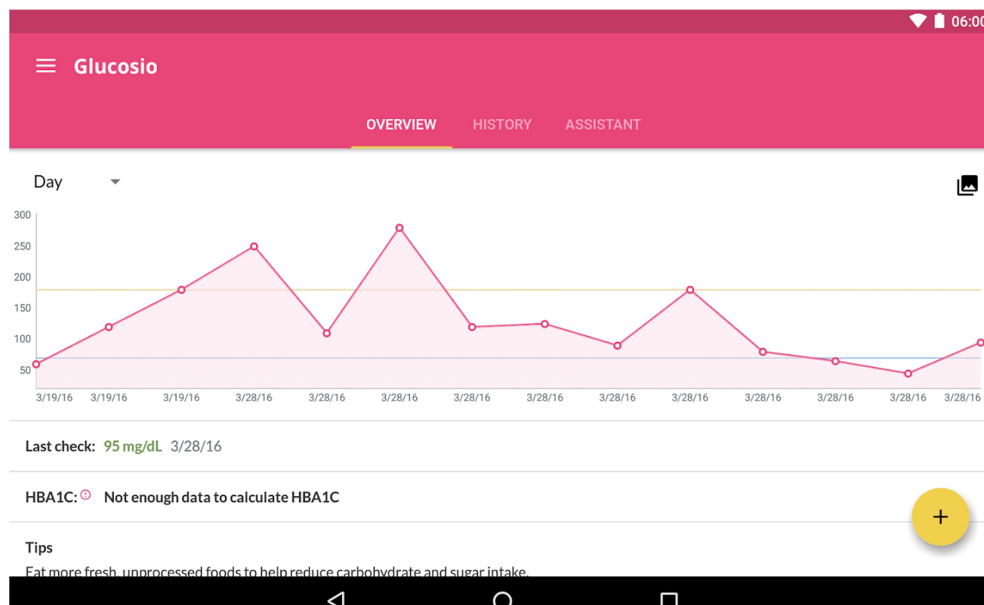
dizajn Glooko platforme pokazuju veliku primjenjivost u liječenju i prevenciji dijabetesa, a neke zdravstvene ustanove u SAD-u ostvarile su suradnju s Glooko kompanijom te započele s eksperimentalnom fazom. S obzirom da je Glooko komercijalan proizvod, za korištenje je potrebna godišnja pretplata [29].



Slika 2.9. Glooko web i mobilno programsko rješenje za liječnike i pacijente [29].

Glucosio, besplatan projekt otvorenog koda namijenjen istraživanju i upravljanju dijabetesom, nastao je zahvaljujući grupi razvojnih programera koji su htjeli razviti programsku podršku koja će istovremeno oboljelima pomoći u praćenju i samoupravljanju dijabetesom, a znanstvenicima i inženjerima u istraživanju i pronalaženju što učinkovitijih načina u ICT području za borbu protiv dijabetesa. Glucosio se sastoji od Android i iOS aplikacije pomoću kojih oboljeli mogu pratiti svoje zdravstveno stanje te povezati ostale usluge i aplikacije (poput MyFitness Pal aplikacije za praćenje fizičke aktivnosti) kako bi se dobio bolji pogled na praćenje dijabetesa. Korisnici mogu pomoći znanstvenicima, inženjerima i zdravstvenim djelatnicima na način da dopuste anonimno korištenje njihovih podataka (mjerenja, praćenja, upravljanja). S obzirom da je projekt u potpunosti otvoren, otvoren je i pristup programskom rješenju na GitHubu (sustavu za upravljanje programskim kodom i suradnju) te znanstvenici i inženjeri

moгу sudjelovati i pomoći u razvoju ove platforme ili ju koristiti u stvaranju vlastitih usluga [30].

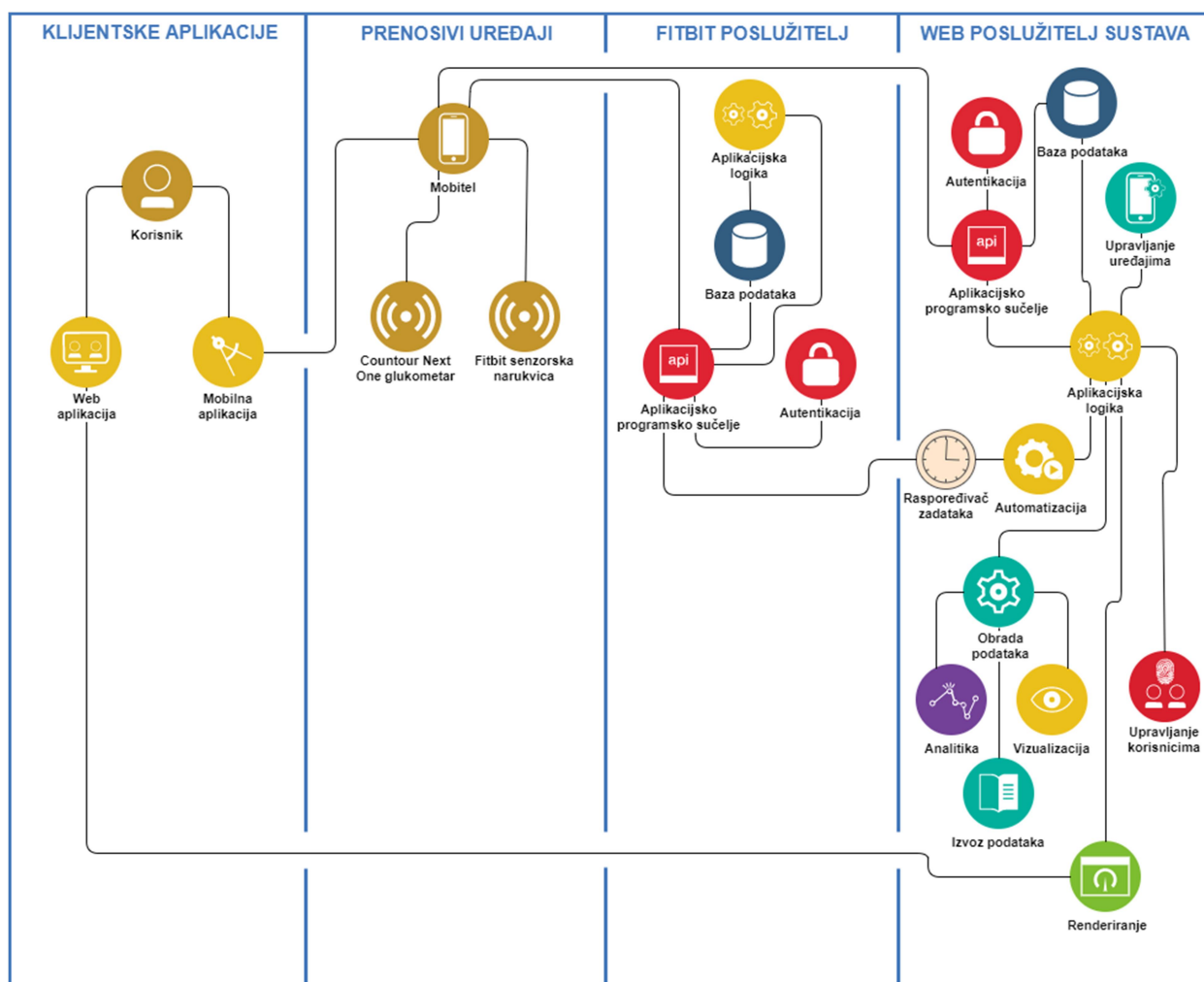


Slika 2.10. Dio Glucosio mobilne aplikacije koja je dio projekta namijenjenog istraživanju i upravljanju dijabetesom [30].

Postoji velik broj programskih rješenja u obliku web baziranih sustava ili platformi i samostalnih aplikacija poput spomenutih za upravljanje, praćenje, dijagnostiku i liječenje dijabetesa te su nova rješenja u stalnom porastu, međutim većina njih ne uključuje važne funkcije ili nudi tek nekoliko funkcija na koje su koncentrirane. Neke nisu jednostavne za korištenje, kvaliteta velike većine je upitna zbog nedostatka suradnje sa stručnjacima iz područja zdravstva, a vrlo je malen broj onih koje su prihvaćene od strane zdravstva i integrirane u veće zdravstvene sustave. Istraživanja pokazuju da budući sustavi, usluge i m-zdravlje aplikacije moraju biti osobnije orijentirane, poboljšane u pogledu upotrebljivosti i pristupačnosti te prihvaćene i podržane od strane zdravstvenih službi i institucija [26].

3. PROGRAMSKO RJEŠENJE – SUSTAV ZA PRAĆENJE ZDRAVSTVENOG STANJA OBOLJELIH OD DIJABETESA

Sustav namijenjen praćenju zdravstvenog stanja oboljelih od dijabetesa napravljen u sklopu ovog rada temelji se na potebama, funkcionalnostima i karakteristikama koje su se kroz prethodno diskutirana znanstvena istraživanja pokazale nužnima za implementaciju kvalitetnog i pristupačnog programskog rješenja za poboljšanje zdravstvenih usluga u aspektu dijabetesa. Programsko rješenje, odnosno sustav, sastoji se od nekoliko dijelova: web aplikacije koju koriste liječnici i pacijenti za detaljan pregled zdravstvenog stanja na temelju prikupljenih podataka o mjerenjima i raznih analiza, Android mobilne aplikacije koja nudi korisne funkcije orijentirane prema pacijentu te od programske podrške na strani poslužitelja izrađene na LAMP platformi uz Laravel razvojni okvir zadužene za obradu, manipulaciju i pohranu podataka te primanje zahtjeva i vraćanje odgovora (API) strani klijenta koja može biti web aplikacija ili mobilna aplikacija. Automatizacija prikupljanja podataka ostvarena je intgriranjem Fitbit *fitness* senzorske narukvice te Countour Next One glukometra u sustav. Jedan od najvećih ciljeva sustava je, osim pružanja mogućnosti naprednog pristupa u praćenju zdravstvenog stanja, u što većoj mjeri ukloniti potrebu za ručnim unosom podataka te u što većoj mjeri olakšati pristup podacima automatskim prikupljanjem primjenom suvremenih informacijskih i komunikacijskih tehnologija.



Slika 3.1. Arhitekturalni pregled cjelokupnog programskog rješenja

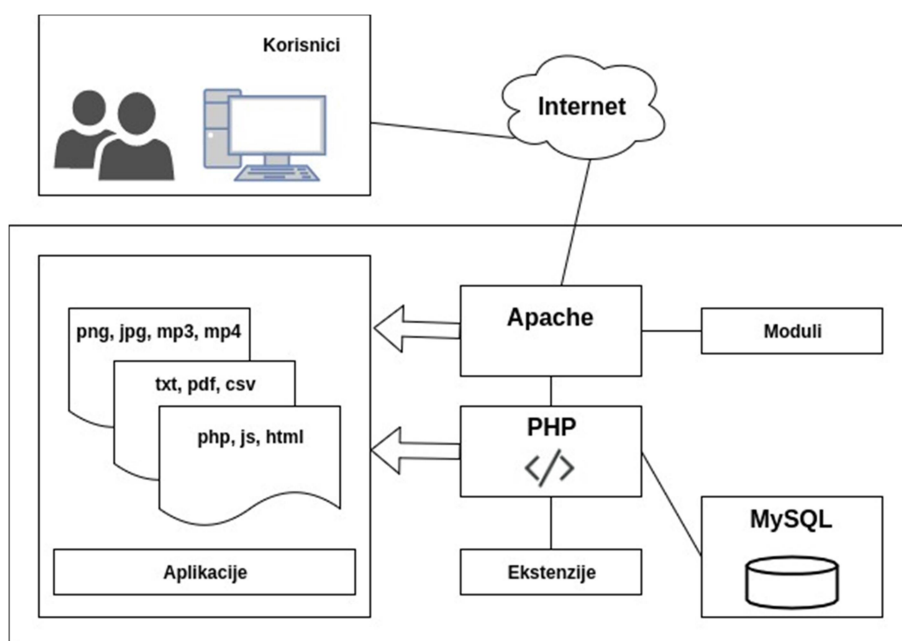
3.1. Opis platformi, alata i tehnologija

Glavni dio sustava namjenjenog praćenju zdravstvenog stanja oboljelih od dijabetesa razvijen je na temelju LAMP platforme i Laravel MVC razvojnog okvira. Funkcije sustava proširene su uvođenjem klijentskih aplikacija poput web aplikacije koja se pokreće iz web preglednika te mobilne Android aplikacije koja pacijentima pruža najbitnije funkcije. Pored iskorištavanja standardnih mogućnosti pojedine tehnologije, uvedene su različite biblioteke i proširenja koja povećavaju djelotvornost sustava. U sljedećim poglavljima opisane su najbitnije tehnologije, alati i platforme korištene za razvoj ovog sustava.

3.1.1. LAMP stog

LAMP stog (engl. *Linux, Apache, MySQL, PHP stack*, LAMP) je platforma koja koristi četiri slojevito strukturirane komponente koje zajedno pružaju temelj za izradu vlastite programske podrške na strani poslužitelja: Linux operacijski sustav, Apache HTTP (engl. *Hypertext Transfer Protocol*, HTTP) poslužitelj, MySQL baza podataka i PHP programski jezik

koji je zamjenjiv jezicima Python i Perl. Linux kao operacijski sustav predstavlja najniži sloj te je na njemu smješten Apache kao HTTP web poslužitelj. Apache pruža siguran i učinkovit prijenos podataka putem HTTP protokola te kroz module omogućuje proširivost funkcija poput uvođenja sigurnosne autentikacije, digitalnih certifikata, prilagođavanja poruka o greškama, definiranja višestrukih virtualnih domaćina (engl. *virtual host*), itd. MySQL kao relacijska baza podataka pruža skladište podataka unutar LAMP platforme. Dohvaćanje, formatiranje i obrada podataka u bazi podataka na tradicionalni način moguća je kroz izradu upita koristeći SQL jezik. PHP kao programski jezik predstavlja najviši sloj strukture LAMP stoga te programeru omogućuje interakciju s ostalim dijelovima platforme poput pristupa podacima, rad sa HTTP zahtjevima i odgovorima, i pokretanja naredbi na razini operacijskog sustava.



Slika 3.2. Struktura LAMP platforme [31].

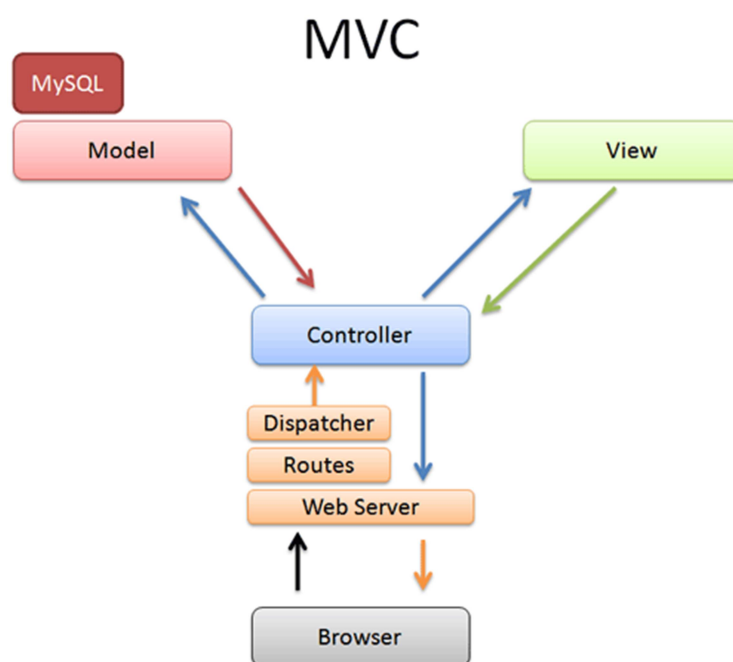
LAMP pruža alternativu komercijalnim paketima i radi kao skup softverskih programa koji pružaju potrebnu platformu za razvoj i implementaciju programskih rješenja na internetu. Nudi jedan od najučinkovitijih i najpopularnijih načina razvijanja web aplikacije kao programskog rješenja cjelokupnog područja djelovanja organizacija (engl. *enterprise solution*) zbog svoje fleksibilnosti, prilagodbe i ekonomičnosti. LAMP platforma predstavlja tehnologiju otvorenog koda visoke razine sigurnosti te pruža mogućnost odabira između različitih distribucija Linux operacijskog sustava poput CentOS, Ubuntu i Debian distribucija [31].

3.1.2. Laravel MVC razvojni okvir

Laravel je moderni web razvojni okvir napisan u PHP programskom jeziku s ciljem poboljšanja kvalitete izrade programske podrške na strani poslužitelja te koristi koncept MVC

(engl. *model-view-controller*) arhitekturnog obrasca. MVC arhitekturni obrazac omogućuje odvajanje programskog koda u logičke cjeline, a to su modeli (engl. *model*), pogledi (engl. *view*) i kontroleri (engl. *controller*). Modeli su klase zadužene za definiranje struktura podataka, entiteta u bazi podataka te interakciju između entiteta. Odgovorni su za rad s podacima, definiranje relacija između podataka i implementaciju poslovnih pravila te pružaju lakše održavanje programskog koda uz smanjenje mogućnosti pojavljivanja grešaka i uz ponovnu upotrebu određenih dijelova koda. Kontroleri sadrže metode koje definiraju logiku aplikacije i pružaju različite funkcionalnosti sustava, upravljaju korisničkim zahtjevima, reagiraju na njih te vraćaju odgovore na zahtjeve. Kada kontroler primi zahtjev (engl. *request*) vrši validaciju podataka, poziva određen model, priprema podatke i nakon toga vraća odgovor. Pogledi (engl. *view*) uzimaju podatke od objekta kontrolera i prosljeđuju pregledniku. Zaduženi su za prikaz podataka i dodaju funkcionalnosti koje su potpuno izolirane od složenih operacija nad podacima. Kod interakcije sa sustavom, pogledi su najčešće predstavljeni kroz HTML. Prednosti korištenja MVC tehnologije su sljedeći [32]:

- lakše održavanje, testiranje te nadograđivanje aplikacije ili poslužiteljske programske podrške
- fleksibilnost kod planiranja i implementiranja objekata i baze podataka
- višestruka iskoristivost i modularnost
- proširivost i skalabilnost



Slika 3.3. Dijelovi MVC arhitekture i međusobne veze između njih [32].

Laravel je jedan od rijetkih PHP razvojnih okvira koji pruža stvarnu modularnost programskog koda. To ostvaruje kroz fleksibilnu mogućnost upravljanja keširanjem (engl. *cache*), sjednicama (engl. *session*), bazama podataka (engl. *database*) i funkcijama autentikacije (engl. *authentication*). Kroz uključivanje dodatnih programskih paketa omogućuje proširenje funkcija samog razvojnog okvira, a s druge strane omogućuje i pakiranje (engl. *bundling*) vlastitih implementacija i rješenja u samostalne programske pakete koji se mogu ponovno koristiti u drugim aplikacijama i sustavima. Kroz takozvane migracije (engl. *migrations*) pruža mogućnosti izrade shema baza podataka na platformski neovisan način što znači da generiranje sheme baze podataka kroz migracije može biti ostvareno na različitim vrstama baza podataka poput MySQL, PostgreSQL, MSSQL i SQLite baza. Vrlo bitna značajka Laravela je Eloquent implementacija objektno relacijskog preslikavanja (ORM, engl. *object-relational mapping*). ORM je tehnika koja u sustavima s relacijskim bazama podataka povezuje objekte aplikacije definirane kroz modele s tablicama baze podataka. Time omogućuje lakše održavanje programske podrške te fleksibilnost u obradi i upravljanju podacima. Laravel kroz Eloquent pruža moderan način interakcije s bazom podataka na objektno orijentiran način gdje se podacima pristupa preko objekta modela koji definira skup podataka predstavljen tablicom u bazi podataka. Umjesto pisanja SQL upita, upiti se korištenjem Eloquent tehnike prema bazi podataka ostvaruju pozivima metoda koje se prema potrebi mogu nadovezivati jedna na drugu, a kao rezultat vraća se kolekcija objekata na kojima je moguće raditi dodatnu obradu. Značajka koja se dodatno ističe je Artisan sučelje naredbenog retka (engl. *command-line interface*). Koristeći Artisan, razvojni programer može automatizirati postupke tijekom razvoja programske podrške poput generiranja klasa modela i kontrolera, dok s druge strane može pokretati migracije nad bazom podataka te pokretati zakazane zadatke i naredbe na poslužitelju [33].

3.1.3. JavaScript, HTML, CSS i Sass

JavaScript je platformski neovisan objektno orijentirani programski jezik koji se unutar okruženja u kojem se pokreće (poput web preglednika) može povezati s objektima i resursima tog okruženja i njima upravljati programskim putem. JavaScript sadrži standardnu biblioteku objekata poput objekata Array, Date, Math, a funkcionalnost je proširena za različite svrhe dopunjavanjem s dodatnim objektima poput objekata strane klijenta (engl. *client-side*) za upravljanjem preglednikom i njegovim objektom modela dokumenta (DOM, engl. *Document Object Model*). Tako na primjer proširenja na strani klijenta omogućuju klijentskoj aplikaciji postavljanje elemenata u HTML strukturu i odgovaranje na različite akcije i događaje poput reagiranja na ulazne podatke (npr. tipkovnica, miš, reakcija na dodir zaslona mobitela). S druge

strane, proširenjima za pokretanje JavaScripta na strani poslužitelja omogućuje se stvaranje poslužiteljske programske podrške u ovom programskom jeziku gdje se pruža izravna komunikacija s bazom podataka, kontinuirano primanje HTTP zahtjeva i vraćanje odgovora, formatiranje podataka i renderiranje pogleda, rad s tehnologijama za prijenos podataka i komunikaciju u stvarnom vremenu poput web socketa ili rad s datotekama na poslužitelju [34].

3.1.3.1. Paketi jQuery, Handlebars, Highcharts, Underscore

Postoje mnogobrojni paketi (biblioteke) razvijeni za JavaScript programski jezik koji pružaju mogućnosti jednostavnijeg razvoja značajki i funkcija web aplikacija. Tako su u ovom sustavu namjenjenom za praćenje zdravstvenog stanja iskorišteni paketi poput jQueryja, Handlebarsa, Highchartsa i Underscorea.

jQuery pruža opći sloj za apstrakciju pisanja JavaScript programskog koda gdje objedinjuje izvorne funkcije JavaScripta u vlastite metode i time omogućuje lakšu i bržu primjenu ovog programskog jezika u sljedećim situacijama [35]:

- HTML i DOM manipulacija

Podrazumijeva upravljanje elementima korisničkog sučelja, jednostavniji način pristupanja elementim, pronalaženje željenih dijelova i struktura te efikasan način dohvaćanja i manipulacije nad podacima.

```
$('#div.content').find('p');
```

- manipulacija CSS stilskim svojstvima, animiranje elemenata korisničkog sučelja

```
$('#div.details').slideDown();
```

- upravljanje događajima i akcijama korisnika

```
$('#button.show-details').click(function() { $('#div.details').show(); });
```

- AJAX (engl. *Asynchronous JavaScript and XML*) - asinkrono slanje zahtjeva na određen događaj ili poziv metoda sa strane klijenta i primanje odgovora od strane poslužitelja

```
$.get(APP_URL + '/admin/patients/' + patientId, function(r) {}, "json")
    .done(function(data) {
        console.info(data);
    })
    .fail(function (data) {
        console.log(data);
    });
```

- pojednostavljenje čestih zadataka poput dohvaćanja i manipulacije objekata iz polja (engl. *arrays*) te parsiranja JSON transportnog formata podataka

```
$.each(dataObj.maxs, function(key, value) {
    value.y = (value.y/1000)/60;
});
```

Jedna od najkorisnijih biblioteka za rad s različitim vrstama podataka, manipulaciju i obradu u JavaScript programskom jeziku je Underscore. Sastoji se od više od stotinu funkcija koje spadaju u četiri glavne kategorije ovisno o vrstama podataka kojima manipuliraju: funkcije za manipulaciju polja podataka, funkcije za manipulaciju objekata, funkcije za manipulaciju i polja i objekata i funkcije za manipulaciju ostalih vrsta podataka. Te funkcije znatno olakšavaju najčešće potrebe u smislu obrade i formatiranja skupova različitih vrsta podataka. Neke često korištene funkcije su sljedeće:

- **map** - stvara novo polje vrijednosti dobiveno mapiranjem svake vrijednosti u danom polju kroz transformacijsku funkciju:

```
_.map([1, 2, 3], function(num) { return num * 3; });
=> [3, 6, 9]
```

- **filter** - vraća polje svih vrijednosti koje prolaze provjeru istinitosti u funkciji:

```
_.filter([1, 2, 3, 4, 5, 6], function(num) { return num % 2 == 0; });
=> [2, 4, 6]
```

- **pluck** - izvod vrijednosti iz polja na temelju dane ključne vrijednosti:

```
_.pluck([{name: 'moe', age: 40}, {name: 'curly', age: 60}], 'name');
=> ["moe", "curly"]
```

- **min** - vraća najmanju vrijednost iz danog polja:

```
_.min([10, 5, 100, 2, 1000]);
=> 2
```

Također, Underscore pruža i funkcije za jednostavnu provjeru vrste podatka i postojanosti podataka poput `_.isArray(val)`, `_.isObject(val)`, `_.isString(val)`, `_.isNumber(val)`, `_.isNull(val)`, `_.isUndefined(val)`.

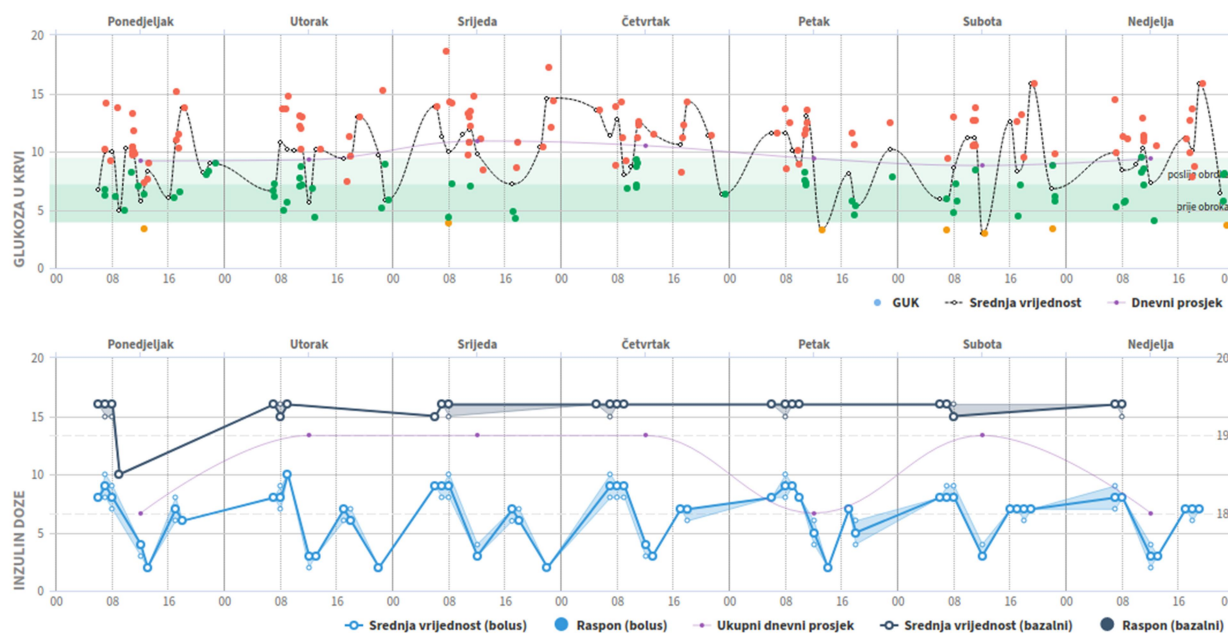
Handlebars je poznat JavaScript *template engine* koji se koristi na strani klijenta za potrebe dinamičkog generiranja prikaza korisničkog sučelja, odnosno HTML-a, na temelju ulaznih podataka i definiranog predloška (engl. *template*). Koristi se u slučajevima kada su prikazi sučelja aplikacije često promjenjivi ili se često osvježavaju posebice kao rezultat

promjena u skupovima podataka dohvaćenih od strane REST API-ja ili neke druge vrste poslužitelja prema strani klijenta, kada je aplikacija vrlo responzivna na akcije korisnika ili kada se razvija jednostranična web aplikacija (SPA, engl. *single-page application*). Zahvaljujući Handlebarsu, prilikom punjenja korisničkog sučelja podacima iz formata podataka poput JSON formata, generiranje prikaza tih podataka u korisničkom sučelju potpuno je izolirano i odvojeno od logike JavaScript programskog koda te se u predlošku koji treba prikazati podatke definiraju izrazi s varijablama koje će zamijeniti stvarni podaci. Handlebars je prevoditelj te generira HTML tako što se njegovoj funkciji predaju željeni podaci kao objekt (ili red objekata) nakon čega vraća string strukture HTML-a sa svim svojstvima objekta uključenim na relevantnim mjestima predloška [36]. Jednostavan primjer primjene Handlebars biblioteke iz web aplikacije sustava:

```
JavaScript:
...
function(data) {
    var template = $('#bg-summary-hb').html();
    $('#bg-summary-hb-holder').html(Handlebars.compile(template)(data));
}
...
HTML prezložak:
...
<div class="label label-info">
    <!-- total_logs atribut objekta data -->
    <b>{{total_logs}}</b> mjerenja ukupno
</div>
<div class="label label-info">
    <!-- avg_logs_per_day atribut objekta data -->
    <b>{{avg_logs_per_day}}</b> mjerenja/dan
</div>
...
```

Highcharts je jedna od najpoznatijih JavaScript biblioteka koja se koristi u implementiranju naprednih vizualizacija podataka. Nudi programsku podršku za izradu intuitivnih i interaktivnih vektorskih grafikona različitih vrsta: linijski, površinski, stupčasti, trakasti, tortni, polarni, te mnogo drugih uz mogućnost kombiniranja različitih vrsta grafikona. Highcharts pruža vrlo napredne mogućnosti u vizualizaciji podataka uz visoku razinu prilagodbe vlastitim potrebama. Također, pruža reponzivnost interaktivnih grafikona što znači da je omogućena adaptacija prema uređajima manjih dimenzija poput tableta ili mobitela. Prihvaća

podatke u različitim formatima poput JSON i CSV formata te omogućuje učitavanje i ažuriranje podataka uživo, odnosno u bilo kojem trenutku nakon početne inicijalizacije i renderiranja.



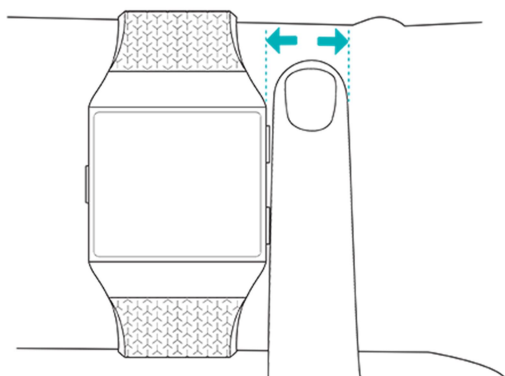
Slika 3.4. Neki grafikoni implementirani koristeći Highcharts biblioteku.

3.1.4. Fitbit i OAuth 2.0

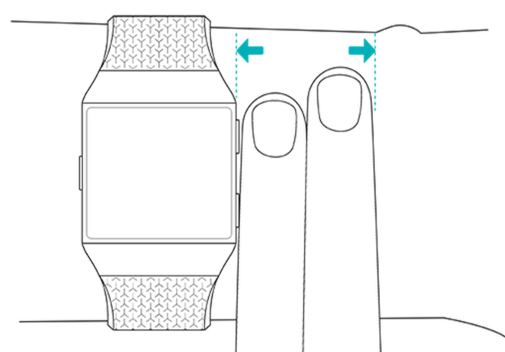
Fitbit Charge HR je prenosivi senzorski uređaj u obliku narukvice za neinvazivno mjerenje aktivnosti. Nudi mjerenje i praćenje rada srca (puls), broja napravljenih koraka, prijeđene udaljenosti, broja popetih katova, fizički aktivnih minuta, fizičke aktivnosti, potrošenih kalorija te duljinu i karakteristike sna. Pored praćenja aktivnosti, Fitbit kao organizacija nudi web i mobilnu aplikaciju za praćenje i pregled tjelesne aktivnosti te su te aplikacije okrenute i koncentrirane prema sportu i fitnessu. Istraživanja u 2014. godini napravljena na temelju osam Fitbit narukvica u trajanju od 70-ak minuta tjelovježbe uključujući 13 različitih aktivnosti, pokazala su da su ove senzorske narukvice “razumno točne” te su pokazivale greške od desetak posto. Novija istraživanja u 2015. godini pokazala su bolje rezultate te su odstupanja od stvarnih vrijednosti iznosila od šest do deset posto.

Fitbit Charge HR uređaj koji je korišten u sklopu praktičnog dijela ovog rada omogućio je prikupljanje podataka u stvarnom vremenu te mjerenje i prikupljanje vrijednosti podataka pacijenta o puls, koracima, fizičkoj aktivnosti i intenzitetu, potrošenim kalorijama te snu. Fitbit koristi vlastitu PurePulse tehnologiju za automatsko i kontinuirano praćenje pulsa tijekom cijelog dana. Kada srce kuca, kapilare se proširuju na temelju promjene volumena krvi. Fitbitova PurePulse LED svjetla reflektiraju se na kožu kako bi senzorski uređaj detektirao promjene volumena te se primjenjuju prilagođeni algoritmi za kontinuirano mjerenje pulsa. Intenzitet rada

srca koristi se gotovo u svim aspektima mjerenja. Iz tog razloga bitno je što duže nositi ovu senzorsku narukvicu jer što više podataka kroz vrijeme prikupi, mjerenja će biti točnija. Veliku ulogu u tome ima intenzitet srca u stanju mirovanja (engl. *resting heart rate*) koji uređaj neprekidno prikuplja u trenucima mirovanja kada je osoba budna ili spava. Za točnija očitavanja bitno je naglasiti ispravnost nošenja senzorske narukvice na ruci pa je iz tog razloga za vrijeme mirovanja ili za vrijeme ne bavljenja fizičkom aktivnošću preporučeno nositi uređaj kao na slici 3.5.a, a za vrijeme fizičke aktivnosti kao na slici 3.5.b.



Slika 3.5.a. Položaj senzorske narukvice za vrijeme mirovanja [38].



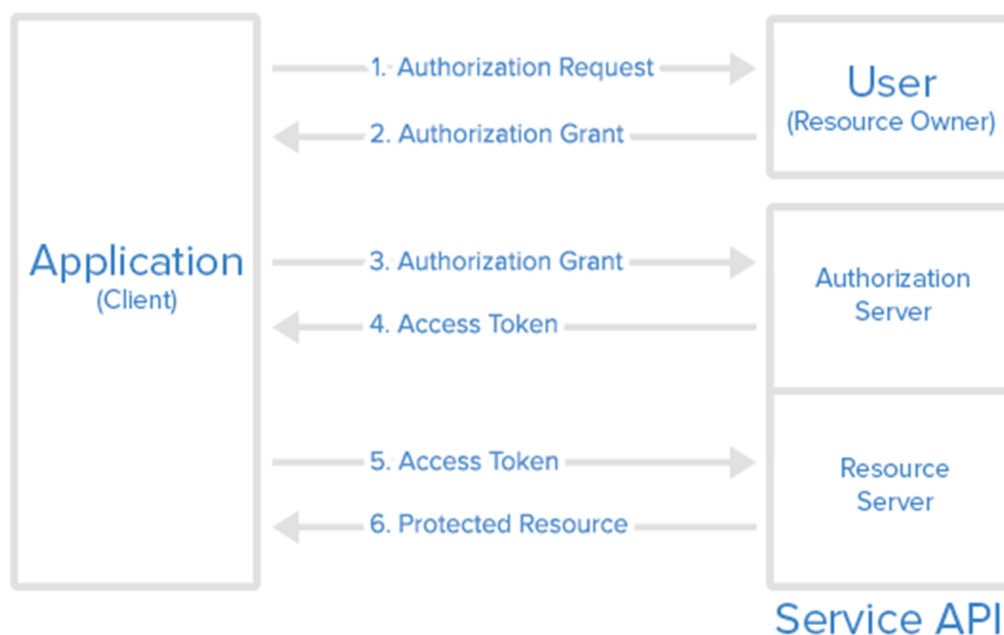
Slika 3.5.b. Položaj senzorske narukvice za vrijeme trajanja fizičke aktivnosti [38].

Za mjerenje koraka, Fitbit uređaji koriste 3-osni akcelerometar za detektiranje pokreta. Akcelerometar je elektromehanički uređaj koji mjeri sile nastale akceleracijom ili ubrzanjem. Te sile mogu biti statične, kao stalna sila gravitacije ili dinamične - uzrokovane kretanjem ili vibriranjem akcelerometra. Ovisno o tipu, može mjeriti sile nastale akceleracijom na jednoj, dvije ili tri osi koristeći kapacitivne pločice na temelju promatranja promjene kapaciteta. Djelovanjem sile u jednom smjeru kapacitet se mijenja te je moguće detektirati akceleraciju u pojedinom smjeru [37]. Fitbit analizom podataka o ubrzanju pruža detaljne informacije o učestalosti, trajanju, intenzitetu i uzorcima kretanja te određuje broj napravljenih koraka, prijeđen put, potrošene kalorije i karakteristike i kvalitetu sna. Također, u pogledu praćenja broja koraka, Fitbit senzorske narukvice koriste optimizirane algoritme za brojanje koraka gdje se traže obrasci gibanja koji su najizrazitiji kod većine ljudi kada hodaju. Algoritmi određuju je li količina pokreta dovoljno velika postavljanjem pragova. Ako pokret i naknadno mjerenje ubrzanja zadovoljavaju uvjete praga, pokret se računa kao korak. Automatizirano mjerenje fizičke aktivnosti i intenziteta realizirano je kroz Fitbitovu SmartTrack tehnologiju koja omogućuje otkrivanje kontinuiranih aktivnosti poput hodanja, trčanja i bicikliranja, a kod nekih novijih uređaja omogućeno je i otkrivanje aktivnosti poput plivanja i aerobnih aktivnosti (poput

Zumbe, kardio-kickboxinga, plesa, itd.). SmartTrack ima ulogu u prepoznavanju aktivnih minuta gdje je aktivnost koja se obavlja znatno napornija od analiziranih podataka u stanju mirovanja ili regularnog hodanja, a to uključuje sve aktivnosti od brzog hoda do kardio treninga ili trčanja. Osim u praćenju pokreta i kretanja, velik značaj u otkrivanju ovakvih aktivnosti je u naprednom praćenju rada srca gdje se u obzir uzima intenzitet otkucaja srca. Potrošnja energije (kalorija) računa se standardiziranim putem na temelju starosti, spola i tjelesne mase gdje Fitbit u obzir uzima intenzitet rada srca kroz vrijeme što predstavlja vrlo pouzdan način mjerenja potrošnje kalorija. Automatsko otkrivanje sna temelji se na pokretima, kretanju i uzorcima intenziteta rada srca. Uređaj pretpostavlja san nakon otprilike sat vremena stanja mirovanja gdje u obzir uzima moguće manje pokrete i niži puls od uobičajenog. Fitbit uređaj prati i faze spavanja koje se procijenjuju kombinacijom pokreta i uzoraka rada srca za vrijeme sna. Uređaj prati promjene u brzini otkucaja srca, poznate pod nazivom varijabilnost brzine otkucaja srca (HRV, engl. *heart rate variability*), koje se mijenjaju tijekom prijelaza između laganog sna, dubokog sna i stupnjeva REM faza. S obzirom da se prikupljaju veće količine podataka i koriste se kompleksnije programske rješenja za ove procjene, proračuni se vrše prilikom sinkronizacije senzorskog uređaja s Fitbitovim servisom na web poslužitelju gdje se nalazi odgovarajuća programska podrška [38] [39].

Pristup podacima u stvarnom vremenu omogućen je kroz vlastitu implementaciju programske podrške na temelju OAuth 2.0 standarda. OAuth predstavlja standardizirani autorizacijski okvir koji aplikacijama omogućuje ograničen pristup resursima putem HTTP usluga poslužitelja. Djeluje na način da korisnicima stranih aplikacija omogućuje tok autentikacije kroz web ili desktop aplikacije te aplikacije na mobilnim uređajima. OAuth sadrži jasno definirane uloge, vrste odobrenja autorizacije, slučajeve upotrebe te autentikacijske tokove. Uloge koje definira OAuth standard su: vlasnik resursa (engl. *resource owner*), klijent (engl. *client*) te poslužitelj resursa i autorizacije (engl. *resource server, authorization server*). Vlasnik resursa je korisnik (engl. *user*) koji autorizira stranu aplikaciju da pristupi resursima korisničkog računa. Pristup aplikacije korisničkom računu je ograničen kroz takozvani *scope* nad kojim je dozvoljena autorizacija (primjerice *read* ili *write* pristup). Poslužitelj resursa sadrži zaštićene korisničke račune, a njegova autorizacijska strana potvrđuje identitet korisnika koji zatražuje pristup resursima te nakon potvrde isporučuje *access token* stranoj aplikaciji. Sa stajališta razvojnog programera strane aplikacije, API usluge poslužitelja koji koristi OAuth objedinjuje uloge poslužitelja resursa i autorizacije zajedno u jednu cjelinu. Klijent je predstavljen stranom aplikacijom koja traži pristup resursima korisničkog računa na strani poslužitelja domaćina. Prije

nego ostvari pristup, mora biti odobrena kroz autorizaciju i validaciju od strane API-ja poslužitelja.



Slika 3.6. Pojednostavljen tok autorizacije OAuth 2.0 standarda [40].

Pojašnjenje toka autorizacije OAuth 2.0 standarda:

1. strana aplikacija šalje zahtjev za autorizacijom pristupa resursima korisnika
2. korisnik autorizira zahtjev i API poslužitelja vraća odobrenje autorizacije
3. strana aplikacija šalje zahtjev za dohvaćanje *access tokena* od strane autorizacijskog poslužitelja (putem API-ja) uz prilaganje informacija o vlastitom identitetu i primljenom odobrenju autorizacije
4. ukoliko je identitet prihvaćen i autenticiran, autorizacijski poslužitelj (API) izdaje *access token* aplikaciji koja je zatražila zahtjev
5. strana aplikacija šalje zahtjev za dohvaćanje resursa od poslužitelja resursa (API) uz prilaganje *access token* podatka za autentikaciju
6. ukoliko je *access token* ispravan, poslužitelj vraća traženi resurs stranoj aplikaciji

Postoji više načina pristupa resursima poslužitelja koji koristi OAuth 2.0 standard te se razlikuju po tipu odobrenja autorizacije (engl. *Authorization Grant Type*). Svaki pristup sadrži vlastite modifikacije standardnog toka autorizacije gdje neki pojednostavljuju tok, a neki dodaju dodatne zahtjeve u standardni OAuth tok [40].

U sustavu realiziranom u praktičnom dijelu rada korišten je *Authorization Code* tip odobrenja autorizacije. Ovaj tip je najrasprostranjeniji tip autorizacije korišten kroz OAuth 2.0

standard jer je optimiziran za aplikacije na strani poslužitelja (engl. *server-side applications*). Modificirani tok ovog tipa dodaje dodatnu vrstu tokena naziva *refresh token*. Ovaj token uvodi se iz sigurnosnih razloga jer token za pristup (*access token*) ima ograničeno vrijeme trajanja te se nakon njegovog isteka treba zatražiti novi slanjem zahtjeva s prethodno primljenim *refresh tokenom*. Postupak inicijalnog postavljanja, autoriziranja i povezivanja Fitbit senzorskog uređaja u razvijeni sustav te primarno ostvarivanje komunikacije u stvarnom vremenu između vlastitog sustava i Fitbitovih usluga ostvareno je implementacijom *Authorization Code Grant Flow* pristupa koji se sastoji od sljedećih koraka [41]:

1. sustav preusmjerava korisnika na službenu Fitbit autorizacijsku stranicu gdje korisnik potvrđuje autorizaciju sustava
2. Fitbit preusmjerava natrag u sustav zajedno sa priloženim autorizacijskim kodom (engl. *authorization code*)
3. sustav šalje zahtjev prema Fitbit API-ju za izdavanje *access tokena* i *refresh tokena* uz prilaganje autorizacijskog koda
4. sustav sprema *access token* i *refresh token* te ih koristi za automatizirano prikupljanje podataka od usluge Fitbita (API)
5. nakon detekcije isteka tokena za pristup resursima, automatiziranim putem osvježava se *access token* slanjem zahtjeva uz prilaganje *refresh tokena* kako bi se mogli dohvaćati novi podaci

3.1.5. Android

Android je mobilni operacijski sustav koji se temelji na modificiranoj inačici Linux operacijskog sustava. Izvorno je razvijen kroz *startup* istog naziva, Android Inc. 2005. te ga je nedugo nakon toga Google kupio i preuzeo svu dotadašnju razvijenu funkcionalnost, isto kao i razvojni tim. S obzirom da je Google želio da Android bude otvoren i besplatan, većina Androidovog izvornog koda objavljena je kroz Apache License licencu otvorenog koda što znači da svatko može pristupiti, proučavati, nadograđivati i prilagođavati izvorni programski kod Android operacijskog sustava ili na temelju postojećeg napraviti vlastiti. Velika prednost u razvijanju aplikacija na Android operacijskom sustavu je ta što Android pruža jedinstven i uniforman pristup razvoju aplikacija. To znači da razvojni programeri mogu iste aplikacije pokretati na brojnim različitim uređajima koji su pokretani Android operacijskim sustavom. Osim najpopularnije kategorije takvih uređaja - mobitela, pametni satovi, kućanski aparati, uređaji u automobilima, kamere, pametni televizori i igrače konzole samo su neke od kategorije uređaja koje također koriste Android kao operacijski sustav.

S obzirom da je Android otvoren i dostupan proizvođačima za prilagodbu, prilikom potrebe za izradom nadogradnji, nema ograničenja u hardverskim ili softverskim konfiguracijama [42]. Međutim, neke od najbitnijih značajke koje Android originalno pruža su:

- korisničko sučelje (UI, engl. *user interface*)
- pohrana podataka - korištenje SQLite kao lagane relacijske baze podataka
- povezivanje - podržava velik spektar mreža i mogućnosti povezivanja poput GSM/EDGE, Bluetooth, LTE i Wi-Fi tehnologija
- razmjena poruka - SMS i MMS, Android Google Cloud Messaging (GCM) servis za razmjenu poruka u stvarnom vremenu putem interneta
- podrška prema različitim multimedijским datotekama poput H.263, H.264, MPEG-4, MP3, MIDI, WAV, JPEG, PNG, GIF i drugih
- *multi-touch* - izvorna podrška za prepoznavanje višestrukih dodira ekrana osjetljivog na dodir istovremeno
- *multi-tasking* - izvođenje više zadataka, aplikacija i procesa istovremeno
- Wi-Fi Direct - tehnologija koja dozvoljava otkrivanje i povezivanje različitih uređaja putem visokopropusne *peer-to-peer* veze
- NFC - tehnologija koja dopušta instantnu razmjenu podataka dodirom dvaju NFC podržana uređaja

Ove značajke zajedničke su većini Android uređaja te predstavljaju skup najbitnijih značajki koje Android pruža. Android omogućuje upravljanje i pristup mnogim resursima uređaja, ovisno o mogućnostima hardvera. Tako na primjer neki uređaji sadrže širok spektar senzora koji su prepoznati od Android operacijskog sustava te je razvojnim programerima omogućen pristup tim resursima i iskorištavanje mogućnosti u razvoju vlastite Android aplikacije. To su primjerice žiroskop, barometar, magnetometar, akcelerometar, senzor udaljenosti, senzor vlage, itd.

Za razvoj aplikacija za Android operacijski sustav poput mobilnih aplikacija, potrebno je podesiti razvojno okruženje i programsku podršku za razvoj aplikacija. Najzastupljenije razvojno okruženje koje se danas koristi je Android Studio koji pruža moderan pristup u programiranju, razvijanju, izgradnji i testiranju Android aplikacija. S obzirom da se u Android programiranju najčešće koristi Java programski jezik, uz Android studio koristi se Java JDK (Java Development Kit) koji se sastoji od Java Runtime Environmenta (JRE) zaduženog za pokretanje aplikacija pisanih u Java programskom jeziku, Java interpretera i kompajlera te dodatnih alata. Prilikom razvoja aplikacija, postoje četiri glavne komponente koje treba razlikovati:

1. Aktivnosti (engl. *Activities*)

- predstavljaju jedinstven zaslon s korisničkim sučeljem u kojem se obavlja neka radnja
 - aplikacija se može sastojati od više aktivnosti te svaka aktivnost ima svoj životni ciklus (engl. *Activity Life Cycle*) kojeg je za efikasan razvoj aplikacije potrebno u potpunosti razumjeti
2. Usluge (engl. *Services*)
- komponente koje se pokreću u pozadini i obavljaju dugoročne operacije poput dohvaćanja podataka (jednokratnog ili periodičnog) od poslužitelja, pokretanje glazbe, itd.
3. *Broadcast Receivers*
- komponente zadužene za pružanje odziva na odaslane poruke od strane sustava ili drugih aplikacija unutar sustava
 - nude mogućnost reagiranja na događaje i pokretanje određenih akcija na pojedini događaj
4. Pružatelji sadržaja (engl. *Content Providers*)
- omogućuju pružanje i prijenos podataka iz jedne aplikacije prema drugima na zahtjev koji se mogu pohraniti u datotečnom sustavu, bazi podataka ili negdje drugdje
 - pružatelji sadržaja implementirani su tako da nasljeđuju klasu *ContentProvider* i moraju implementirati standardni skup metoda koji omogućuju sučelje prema ostalim aplikacijama i razmjenu podataka

Dodatne komponente koje su najzastupljenije u razvoju Android aplikacija su sljedeće:

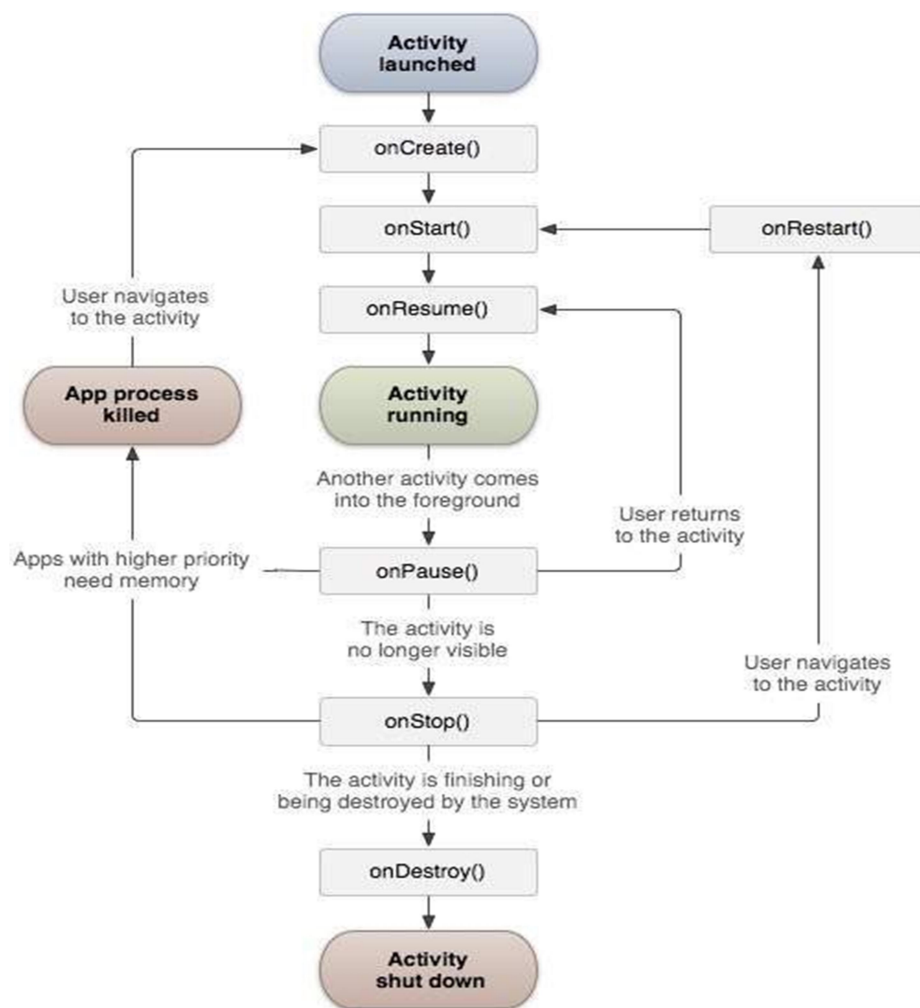
1. Fragmenti (engl. *Fragments*)
 - predstavljaju dio korisničkog sučelja unutar Activity komponente
2. Pogledi (engl. *Views*)
 - UI elementi koji su renderirani u korisničkom sučelju, predstavljeni u programskom kodu kroz objekte poput objekta *Button*, *ListView*, itd.
3. Intenti (engl. *Intents*)
 - objekti koji omogućuju izradu zahtjeva prema dohvaćanju drugih komponenti unutar aplikacije, poput dohvaćanja, prijena podataka i pokretanja druge aktivnosti aplikacije
4. Resursi (engl. *Resources*)

- vanjski elementi aplikacije poput stringova, konstanti, ikona, definicija dimenzija i veličine fontova, itd.

5. Manifest datoteka

- glavna konfiguracijska datoteka svake aplikacije

Svaka aktivnost sadrži metode *onCreate()*, *onStart()*, *onResume()*, *onPause()*, *onStop()*, *onDestroy()*, *onRestart()* te se te metode pozivaju ovisno o stanju aplikacije i predstavljaju životni vijek aktivnosti. Ovisno o stanju kroz koje aktivnost prolazi, programski kod unutar navedenih metoda bit će izvršen. Dijagram koji pokazuje prijelaz u različite faze životnog ciklisa prikazan je na slici 3.7 [43].



Slika 3.7. Životni ciklus aktivnosti i veze između pojedinih faza [43].

3.1.5.1. Retrofit i Gson

Retrofit je popularni REST klijent napravljen za Android i Java programski jezik. Omogućuje uredno i relativno jednostavno dohvaćanje i slanje podataka u nekom od strukturnih formata poput JSON transportnog oblika podataka ili XML oblika putem REST bazirane web

usluge. U razvijenom sustavu na strani Androida koristi se Gson konverter zajedno s Retrofit API klijentom za potrebe serijalizacije i deserijalizacije podataka. Gson je Java biblioteka zadužena za pretvaranje JSON formata podataka u Java objekte i obratno. Velika prednost u korištenju Retrofita zajedno s Gson bibliotekom je mogućnost definiranja hijerarhije podataka i relacija (odnosa među podacima) na razini objekata. Zbog te značajke omogućeno je jedinstveno modeliranje entiteta i resursa, koji se primjerice dohvaćaju s udaljene baze podataka putem REST web API-ja, na isti način kao što su oni definirani na udaljenom poslužitelju u bazi podataka. Pri izradi HTTP zahtjeva Retrofit u pozadini koristi OkHttp biblioteku. Retrofit omogućuje uniformno definiranje adresa odredišta resursa ili krajnjih točaka komunikacije putem HTTP protokola (engl. *endpoint*), a prilikom dohvaćanja podataka osigurava parsiranje u Plain Old Java Object (POJO) objekte čija su svojstva definirana u zasebnim klasama svakog resursa (modela).

Prilikom korištenja funkcionalnosti Retrofita uglavnom je potrebno definirati tri klase:

- model klasu koja se koristi za mapiranje JSON podataka (serijalizacija i deserijalizacija)
- *interface* koji definira moguće HTTP operacije i zahtjeve
- Retrofit.builder klasu koja predstavlja instancu koja koristi sučelje Retrofita za pokretanje HTTP operacija

Svaka metoda *interfacea* predstavlja jedan mogući API poziv. Mora biti naznačena HTTP anotacijom koja definira tip zahtjeva (poput GET ili POST tipa) i relativnu adresu resursa (URI, engl. *uniform resource identifier*):

```
@GET("users")
Call<List<User>> getUsers()
```

Kroz anotacije moguće je koristiti blokove koji će kasnije biti zamijenjeni navedenim parametrom uz pomoć `@Path` anotacije:

```
@GET("users/{name}/commits")
Call<List<Commit>> getCommitsByName(@Path("name") String name)
```

Parametri upita (engl. *query parameters*) mogu biti dodani definiranjem `@Query` anotacije nad parametrom metode te su automatski nadovezani na URL koji se poziva od strane poslužitelja:

```
@GET("users")
Call<User> getUserById(@Query("id") Integer id)
```

Korisna anotacija za definiranje tipa podatka koji se šalje prema API-ju je i `@Body` koja definira objekt kao tip podatka. Na taj način se zapravo u tijelo zahtjeva (engl. *body*) šalje JSON reprezentacija podataka nastala serijalizacijom navedenog objekta, a u pozadini Retrofit osigurava postavljanje zaglavlja HTTP zahtjeva (engl. *header*) na pravilan tip “application/json” kako bi odredište (REST poslužitelj) razumio oblik podataka koji prima [44]:

```
@POST("users")
Call<User> postUser(@Body User user)
```

3.1.5.2. Bluetooth Low Energy

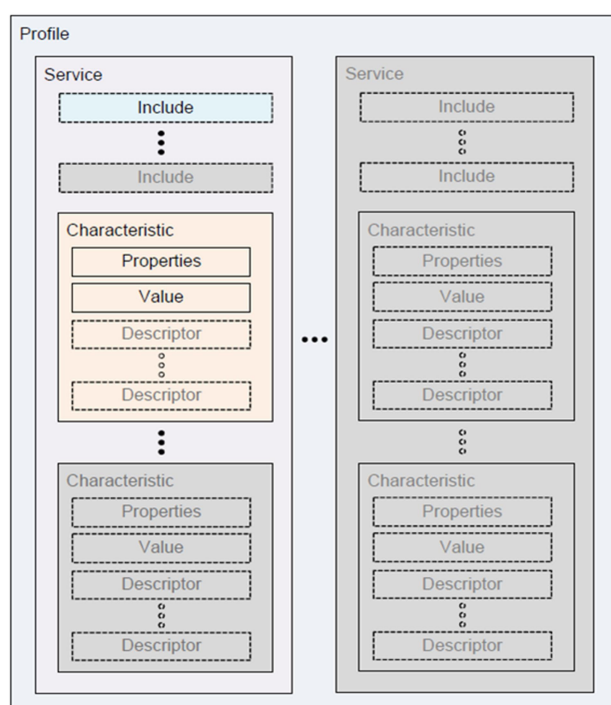
Unutar razvijenog sustava za praćenje zdravstvenog stanja oboljelih od dijabetesa implementirana je mogućnost automatiziranog prikupljanja i sinkronizacije podataka očitavanja razine šećera u krvi između Android aplikacije i Contour Next One glukometra čija je komunikacija u smislu prijenosa podataka temeljena na BLE protokolu. Bluetooth Low Energy ili BLE tehnologija dizajnirana je i osmišljena kako bi pružala funkcionalnost bežičnog prijenosa podataka uz znatno manju potrošnju energije. Takav pristup Android aplikacijama dozvoljava komunikaciju s BLE uređajima koji imaju strože energetske zahtjeve poput glukometara i drugih zdravstvenih uređaja, različitih senzora, automobilskih uređaja i ostalih manjih uređaja. Nasuprot standardnim Bluetooth uređajima, BLE pruža standardiziran pristup u komunikaciji s uređajima ove vrste kroz standardne modele i definicije podataka koji se prenose.

Kako bi dva uređaja mogla međusobno komunicirati putem BLE standarda, BLE uređaji moraju biti u skladu s definiranim pravilima BLE standarda. Ta pravila su poopćena u protokole i konfiguracije [45].

1. Protokol: skup pravila koji definiraju oblikovanje paketa, način usmjeravanja, šifriranje, itd. za razmjenu podataka između dvije strane.
2. Profil: definira način na koji se protokol upotrebljava za postizanje određenih ciljeva. Postoje dvije vrste konfiguracija: generički profili (engl. *generic profiles*) i profili prema upotrebi (engl. *use-case profiles*).
 - a. Generički profili: osnovni i temeljni profili koji su definirani kroz Bluetooth specifikaciju poput GATT profila koji je neophodan u povezivanju i razmjeni podataka između dva uređaja.
 - b. Profili prema upotrebi: profili za specifične slučajeve upotrebe definirani od strane Bluetooth Special Interest Group (SIG) organizacije i ručno od strane proizvođača.

Za potrebe implementacije komunikacije između Android aplikacije i glukometra korišten je GATT profil (Generic Attribute Profile). GATT postavlja pojedinosti i detalje o razmjeni svih profila i korisničkih podataka preko BLE veze. Podaci su organizirani hijerarhijski u odjeljke nazvane usluge (engl. *services*), koje grupiraju konceptijske dijelove podataka nazvanih karakteristike (engl. *characteristic*). Ukratko, podaci koji se razmjenjuju putem BLE standarda su strukturirani podaci, a prijenos podataka putem BLE moguć je samo uz pravilnu implementaciju protokola uz ispravno definiranje usluga i karakteristika koje se pozivaju i izmjenjuju.

Podaci koji se razmjenjuju zovu se atributi, a svaki atribut jedinstveno je identificiran UUID (Universally Unique Identifier) identifikatorom koji predstavlja informaciju u standardiziranoj veličini od 128-bita. Svaki atribut unutar UUID identifikatora sadrži 32-bitni dio koji definira UUID tip (engl. *UUID type*) koji se odnosi na karakteristiku ili uslugu. Atributi koji se prenose putem BLE veze formatirani su kroz karakteristike i usluge. Karakteristika sadrži jedinstvenu logičku vrijednost te može biti promatrana kao tip, analogno klasi. Usluga sadrži kolekciju karakteristika te se preko pojedine usluge pristupa karakteristikama koje ta usluga pruža [46].



Slika 3.8. Struktura i hijerarhija GATT profila [46].

Popis svih usluga i karakteristika, kao i strukture sadržanih podataka i definicija moguće je pronaći na službenoj Bluetooth web stranici.

3.3. Razvoj i način rada sustava za praćenje zdravstvenog stanja oboljelih od dijabetesa

U ovom poglavlju opisana je realizacija cjelokupnog programskog rješenja, odnosno sustava namijenjenog praćenju zdravstvenog stanja oboljelih od dijabetesa. Istaknuti su i objašnjeni dijelovi od kojih se sustav sastoji, opisana je struktura baze podataka koja se koristi i modela koji ju definiraju, objašnjena je implementacija sinkronizacije Fitbit senzorske narukvice i glukometra te su predstavljeni najbitniji dijelovi sustava za krajnjeg korisnika – web i mobilna aplikacija. Bitni segmenti popraćeni su opisom programskog rješenja na strani poslužitelja koje omogućuje funkcionalnost određenih dijelova sustava.

3.3.1. Baza podataka i modeli

Baza podataka koju koristi razvijeni sustav sastoji se od tablica i relevantnih modela prikazanih u tablici 3.1. Također, prikazane su i definirane veze između modela.

Tablica 3.1. Prikaz tablica, modela te odnosa između različitih modela.

DB tablica	Model	Relacije (veze u Laravel MVC okruženju)
base_units	BaseUnit	
roles	Role	User
glucometer_devices_logs	(nema)	
role_user	(nema)	
glucose_logs	GlucoseLog	HealthLogLabel, HealthLog
insulin_log_types	InsulinLogType	InsulinLog
insulin_logs	InsulinLog	InsulinLogType, HealthLogLabel, HealthLog
health_log_labels	HealthLogLabel	
patient_infos	PatientInfo	
health_goals	HealthGoal	User
users	User	FitbitUser, GlucometerDevice, HealthGoal, MedicalTest, PatientInfo, Role
health_logs	HealthLog	User, HealthLogType, GlucometerDevice, CalorieLog, GlucoseLog, InsulinLog, PhysicalActivityLog, SleepLog, StepLog
glucometer_devices	GlucometerDevice	
step_logs	StepLog	HealthLog
calorie_logs	CalorieLog	HealthLog
physical_activity_logs	PhysicalActivityLog	HealthLog
sleep_logs	SleepLog	
health_log_types	HealthLogType	HealthLog
fitbit_users	FitbitUser	FitbitDevice, User
fitbit_devices	FitbitDevice	FitbitUser

Tablica *users* sadrži glavne podatke o korisnicima sustava. Korisnici cjelokupnog sustava su doktori i pacijenti, a podacima se pristupa preko modela User. Osim osnovnih informacija, ovdje je definiran i API ključ za svakog pacijenta koji je generiran prilikom kreiranja

korisničkog računa pacijenta. U tablici *roles* definirane su uloge korisnika (liječnik i pacijent) te je svaki korisnik preko relacijske tablice *role_user* povezan s odgovarajućom ulogom. Model koji je usko vezan uz User model je PatientInfo model s odgovarajućom tablicom *patient_infos*. U toj tablici nalaze se dodatne informacije o korisnicima koji imaju ulogu pacijenta te su s modelom User povezani stranim ključem (engl. *foreign key*) *patient_id*. Svaki zapis u tablici *patient_infos*, odnosno objekt PatientInfo može pripadati jednom korisniku, što je definirano kroz relacijsku metodu *belongsTo()*:

```
public function patient() { return $this->belongsTo(User::class, 'patient_id'); }
```

S druge strane, u generaliziranom modelu pacijenta User, postoji metoda *patientInfo()* koja pozivom vraća objekt PatientInfo, odnosno vraća dodatne informacije o pacijentu:

```
public function patientInfo() {  
    return $this->hasOne(PatientInfo::class, 'patient_id');  
}
```

Ovakva vrsta odnosa između modela u relacijskim bazama podataka naziva se jedan-na-jedan (engl. *one-to-one*).

Tablica *health_goals* sadrži podatke o ciljnim vrijednostima pacijenata. Svaki pacijent može imati više zapisa ciljnih vrijednosti, s obzirom da u stvarnosti liječnik može nakon nekog vremena postaviti nove ciljne vrijednosti. Zbog statističke obrade podataka bitno je čuvati podatke o prošlim ciljnim vrijednostima pa svaki objekt User može imati više objekata (kolekciju objekata) HealthGoal, a svaki objekt HealthGoal može pripadati jednom objektu User. Takva veza između modela User i modela HealthGoal predstavlja jedan-na-više (engl. *one-to-many*) vezu.

Model HealthLog definira tablicu *health_logs* u kojoj se pohranjuju sva mjerenja pacijenata. Neki od atributa koje drži su *value* koji predstavlja vrijednost mjerenja, *log_ts* predstavlja datum i vrijeme mjerenja, *log_type* je strani ključ koji povezuje svako mjerenje s dodatnim informacijama te vrste mjerenja, *insert_type* označava je li mjerenje ručno uneseno ili automatizirano putem uređaja (poput glukometra i Fitbit senzorske narukvice), *user_id* povezuje svako mjerenje s odgovarajućim pacijentom, a *loggable_type* i *loggable_id* odnose se na zapise u polimorfskim tablicama, odnosno u tablicama koje proširuju ovu generalnu tablicu mjerenja. HealthLog model sadrži svojstva koja su zajednička svim mjerenjima, a putem polimorfskih relacija ovaj model povezan je s ostalima koji sadrže dodatne podatke: GlucoseLog, InsulinLog, PhysicalActivityLog, StepLog, CalorieLog i SleepLog. Laravel MVC razvojni okvir kroz

polimorfske relacije pruža generalizaciju modela tako da objekti sličnih atributa zajedničke attribute drže u zajedničkoj tablici koja je u ovom slučaju definirana kroz model `HealthLog`, a vlastite dodatne attribute drže u dodatnoj tablici s pripadajućim objektom modela kojem je relacija s `HealthLog` modelom definirana kroz `morphOne()` metodu. Na primjer, `GlucoseLog` objekt pripada jednom `HealthLog` objektu te je njegova veza definirana na sljedeći način:

```
public function log() { return $this->morphOne(HealthLog::class, 'loggable'); }
```

S druge strane, jedan objekt `HealthLog` također je u ovom primjeru povezan s `GlucoseLog` objektom, ali metodom `morphTo()`:

```
public function loggable() { return $this->morphTo(); }
```

Podrška Laravel razvojnog okvira omogućuje povezivanje s ispravnim polimorfskim modelom na temelju `loggable_type` i atributa kojeg sadrži svaki zapis u tablici `health_logs`. Osim glavnih atributa mjerenja, `GlucoseLog` model sadrži i podatak o oznaci mjerenja kroz relaciju s `HealthLogLabel` objektom, a `InsulinLog` model također na isti način sadrži podatak o oznaci mjerenja, ali i podatak o vrsti inzulina (bazalni ili bolus) kroz relaciju s `InsulinType` modelom. `CalorieLog` i `StepLog` modeli mjerenja nemaju dodatnih atributa, već koriste samo one iz glavnog `HealthLog` modela. `PhysicalActivityLog` model sadrži sljedeće dodatne attribute: `calories`, `steps`, `average_heart_rate` te se za svaki zapis mjerenja fizičke aktivnosti bilježe i ovi podaci. Model `SleepLog` osim atributa iz `HealthLog` modela sadrži i sljedeće: `asleep_mins`, `awake_mins`, `restless_mins`, `sleep_quality`, `awake_count` i `restless_count`.

Dodjeljivanje različitih oznaka mjerenjima kod kojih je to nužno omogućuju relacije tih modela s modelom `HealthLogLabel` koji podatke drži u `health_log_labels` tablici. Kod mjerenja glukoze u krvi i doza inzulina vrlo je bitno pratiti i prije kojeg događaja se odvijalo to mjerenje. Zbog toga su u `health_log_labels` tablici definirane vrijednosti poput „prije doručka“, „poslije doručka“, „prije ručka“, itd. Modeli `GlucoseLog` i `InsulinLog` povezani su s odgovarajućom oznakom kroz relaciju s `HealthLogLabel` modelom:

```
public function label() {  
    return $this->belongsTo(HealthLogLabel::class, 'health_log_label_id');  
}
```

Modeli `GlucometerDevice` i `FitbitDevice` definiraju attribute uređaja za automatizirano prikupljanje podataka. `GlucometerDevice` model povezan je s pacijentom kroz relaciju s modelom `User`, a `FitbitDevice` je sa svakim pacijentom povezan kroz relaciju s `FitbitUser`

modelom koji je u relaciji s User modelom. Svaki objekt FitbitDevice mora biti povezan s odgovarajućim FitbitUser modelom jer su u tablici *fitbit_users* sadržani podaci korišteni za potrebe prethodno spomenutog OAuth 2.0 standarda. I FitbitDevice i GlucometerDevice modeli sadrže attribute bitne za automatiziranu sinkronizaciju podataka s uređajem.

U tablici *base_units* postavljene su standardne vrijednosti jedinica koje se koriste u različitim vrstama mjerenja. Prilikom korištenja mjernih jedinica u sustavu, mjerne jedinice se iščitavaju iz ove tablice preko modela BaseUnit. Tako se na primjer za razinu glukoze u krvi vraća jedinica *mmol/l*, a za primjerice potrošnju kalorija *kcal*.

3.3.2. Sinkronizacija s Fitbit uređajem

Veza između sustava za praćenje zdravstvenog stanja oboljelih od dijabetesa i Fitbit uređaja ostvarena je koristeći koncepte OAuth 2.0 standarda koji je objašnjen u prethodnom tekstu. Podaci koji se prikupljaju su vrijednosti mjerenja fizičkih aktivnosti uz intenzitet rada srca, potrošnje kalorija i kretanje (koraci) te sna.

Ovaj sustav nastoji u što kraćim vremenskim intervalima pristupiti resursima Fitbit uređaja spajajući se na Fitbitov REST OAuth API automatiziranim putem. Kako bi inicijalni pristup bio moguć, pribavljeni su potrebni podaci poput *client_id* i *client_secret* podataka kroz proces registracije serverske aplikacije kroz Fitbitovo korisničko sučelje. Razvijena programska podrška i cijela logika na strani poslužitelja ovog sustava koja omogućuje komunikaciju putem navedenog standarda izolirana je od ostatka programske podrške te je jedino spojena sa sustavom preko modela kako bi se prilikom komunikacije s Fitbit poslužiteljom moglo pristupati internoj bazi podataka. Za izradu HTTP zahtjeva prema udaljenom poslužitelju korištena je GuzzleHttp biblioteka. Implementirano rješenje zadovoljava potrebe sustava te se sastoji od nekoliko komponenata:

- metode za ostvarivanje autorizacije i održavanje komunikacije s Fitbit poslužiteljem
- metode organizirane u klase prema određenom resursu za dohvaćanje relevantnih podataka i vrijednosti mjerenja
- definirane naredbe (engl. *commands*), raspoređivanje zadataka (engl. *task scheduler*) te Cron posla (engl. *Cron Job*)

Metode za ostvarivanje autorizacije i održavanje komunikacije s Fitbit poslužiteljem nalaze se u BaseApiAbstract klasi i Fitbit klasi. BaseApiAbstract klasa sadrži glavna svojstva potrebna za ostvarivanje autorizacije i komunikacije s Fitbit poslužiteljem poput konstanti *BASE_URL*, *CLIENT_ID*, *CLIENT_SECRET*, *TOKEN_EXPIRES*, itd. Implementirane metode

postRetrieveUserAccessData(), *initialFitbitUserSetup()*, *postRefreshUserTokens()* i *getUserAccessTokenForApi()* omogućuju ostvarivanje i održavanje kontinuirane komunikacije s Fitbit poslužiteljem. Inicijalna autorizacija i povezivanje Fitbit uređaja s korisnikom (pacijentom) ovog sustava vrši se pozivima metoda putem objekta Fitbit klase koja nasljeđuje *BaseApiAbstract* klasu. Nakon završetka procesa i dohvaćanja potrebnih podataka, poziva se metoda *initialFitbitUserSetup()* koja stvara novi zapis u bazi podataka tipa *FitbitUser* te sprema sve podatke potrebne za buduću razmjenu podataka s Fitbit poslužiteljem za pacijenta koji je preko *FitbitUser* modela povezan sa svojim uređajem unutar ovog sustava. Metoda *postRefreshUserTokens()* implementirana je za potrebe automatskog obnavljanja pristupnih podataka nakon njihovog isteka: *access_token*, *refresh_token*, *token_expires_in* te *token_timestamp*. Prve dvije vrijednosti bitne su za pristup resursima korisnika na strani Fitbita i zahtjevanje novog pristupnog tokena (*access_tokena*), a druge dvije za mogućnost određivanja valjanosti (isteka) pristupnog tokena:

```
protected function postRefreshUserTokens(FitbitUser $fitbitUser) {
    $tokenRequestTimestamp = \Carbon\Carbon::now();
    $credentials = $this->encodeAppCredentials();
    $accessData = json_decode($this->client->post('oauth2/token', [
        'headers' => [
            'Authorization' => "Basic {$credentials}",
            'Content-Type' => 'application/x-www-form-urlencoded'
        ],
        'form_params' => [
            'grant_type' => self::GRANT_TYPE_REFR_TOKEN,
            'refresh_token' => $fitbitUser->refresh_token,
            'expires_in' => self::TOKEN_EXPIRES
        ]
    ]->getBody());
    $fitbitUser->access_token = $accessData->access_token;
    $fitbitUser->refresh_token = $accessData->refresh_token;
    $fitbitUser->token_expires_in = $accessData->expires_in;
    $fitbitUser->token_timestamp = $tokenRequestTimestamp;
    $fitbitUser->save();
    return $fitbitUser;
}
```

Klase *ActivitiesResource*, *CaloriesResource*, *SleepsResource* i *StepsResource* nasljeđuju osnovne funkcionalnosti nasljeđivanjem *ApiResourceAbstract* klase te implementiraju potrebne metode kako bi objekti ovih klasa mogli koristiti funkcionalnosti nasljeđene klase. *ApiResourceAbstract* sadrži već definirane metode koje sadrže logiku zajedničku klasama

pojednog resursa poput metode *getData(...)*, a kako bi korištenje tih metoda bilo moguće, klasa svakog resursa treba implementirati sljedeće metode *ApiResourceAbstract* klase:

```
abstract class ApiResourceAbstract extends BaseApiAbstract {  
    abstract protected function getResourceModel();  
    abstract protected function getResourceData(  
        FitbitUser $fitbitUser, Carbon $dateFrom);  
    abstract public function updateResourceTable(  
        User $user, $logs, Carbon $dateFrom);  
    ...}
```

Svaka od ovih klasa kroz implementirane metode definira logiku koja se odnosi na resurs kojem objekt svake klase pristupa (poput resursa fizičke aktivnosti ili koraka):

- *getResourceModel()* – vraća model pojedinog resursa (primjerice *StepLog::class*)
- *getResourceData()* – definira logiku dohvaćanja podataka od strane Fitbit poslužitelja
- *updateResourceTable()* – definira logiku sinkronizacije s internom bazom podataka (osvježavanje s novim mjerenjima)

Implementiranjem ovih metoda omogućeno je uniformno pozivanje metoda na objektima različitih klasa prilikom potreba dohvaćanja novih podataka ili sinkroniziranja s bazom podataka sustava.

Laravelov koncept naredbi omogućuje definiranje funkcija koje se mogu pokretati na razini operacijskog sustava iz naredbenog retka. Zajedno s raspoređivačem zadataka i definiranjem Cron posla omogućeno je automatizirano prikupljanje podataka od strane Fitbit poslužitelja. Drugim riječima, čim se Fitbitova senzorska narukvica sinkronizira s Fitbit poslužiteljem, ovaj sustav će sinkronizirati nove prikupljene podatke s bazom podataka. Definirane naredbe (klase tipa *Command*) su:

- *UpdateDeviceInfo*
- *UpdateActivityLogs*
- *UpdateDeviceInfo*
- *UpdateStepLogs*
- *UpdateSleepLogs*

Bitno za napomenuti je da *UpdateDeviceInfo* ima zadatak u periodičkom osvježavanju podataka u vezi korisnikovog *access_tokena* i *refresh_tokena* tako da se osigura ispravnost pozivanja metoda koje dohvaćaju stvarna mjerenja od strane Fitbita. Svaka od ovih klasa u *handle()* metodi

koja izvršava programski kod na poziv naredbe ima definirane metode koje će pozvati. Tako naprimjer UpdateCalorieLogs implementira *handle()* metodu na sljedeći način:

```
$users = User::fitbitUsersWithSetAccess()->get();
foreach($users as $user) {
    $dateFrom = $this->caloriesResource->getDateFromForApiCall($user->fitbitUser);
    $logs = $this->caloriesResource->getData($user, $dateFrom);
    $this->caloriesResource->updateResourceTable($user, $logs, $dateFrom);
}
```

U Kernel.php klasi definirana je logika raspoređivača zadataka kroz *schedule()* metodu. Postavljeno je da se svaka od navedenih naredbi poziva svakih 5 minuta, što iznosi 60 zahtjeva na sat te ne prelazi ograničenja Fitbit API-ja (150). Kako bi bio omogućen ovakav rad u pozadini, definiran je *Cron job* koji predstavlja proces pokrenut na serveru te osigurava pokretanje ovog raspoređivača zadataka:

```
* * * * *
www-data php /home/dir/dest/e-gluko/artisan schedule:run >> fitbit_log.log 2>&1
```

Prva od pet zvjezdica predstavlja minute, druga sate, treća dan u mjesecu, četvrta mjesec i zadnja dan tjedna. Postavljanjem vrijednosti može se definirati koliko često će se pokretati neka naredba u operacijskom sustavu. Znak zvjezdica (*) označava sve moguće brojeve za tu poziciju, pa tako na području „minute“ označava svaku minutu. Drugim riječima, raspoređivač zadataka će se pokretati svake minute i provjeravati postoje li zadaci koje treba obaviti.

3.3.3. Sinkronizacija s glukometrom

Neinvazivno prikupljanje podataka jedan je od najbitnijih faktora u ostvarivanju i razvoju kvalitetno sustava za praćenje zdravstvenog stanja pacijenata. U sustav je uvedena vlastita implementacija automatiziranog postupka prikupljanja podataka s pacijentovog glukometra i sinkronizacija tih podataka s bazom podataka sustava. Zahvaljujući donaciji Contour Next One glukometra i popratne opreme poput lanceta, lancetara i senzorskih trakica za mjerenje razine glukoze u krvi od strane tvrtke MEDiLAB koja se bavi nabavom i distribucijom medicinske opreme, ostvareni su preduvjeti za uvođenje glukometra u sustav. Tijekom testiranja glukometra i rada sa sustavom uzeto je i izmjereno deset uzoraka krvi spomenutim glukometrom.

Contour Next One glukometar koji podržava komunikaciju putem BLE tehnologije unutar sustava izmjenjuje informacije s mobilnom aplikacijom na način kako je to prije objašnjeno, a mobilna aplikacija nakon završetka komunikacije radi zahtjev prema poslužitelju

ovog sustava. Definirana je RESTful ruta, odnosno *endpoint* koji dozvoljava mobilnim aplikacijama pacijenata slanje prikupljenih podataka glukometra:

```
Route::post('glucometer-devices/sync',  
'GlucometerDeviceController@syncGlucometerData');
```

GlucometerDeviceController je kontroler zadužen za rad s udaljenim glukometrom putem Android aplikacije, a metoda koja obrađuje zahtjev sinkronizacije je *syncGlucometerData()*. Metoda prihvata podatke s postavljenim HTTP zaglavljem „application/json“, odnosno JSON format podataka te ih uspoređuje s podacima u bazi. Uspoređuje vremena mjerenja i vremena sinkronizacije te ona mjerenja koja ne postoje pohranjuje u bazu podataka za daljnju analizu i dostupnost pacijentu i liječniku zajedno s ostalim skupovima podataka. Metoda kao odgovor vraća informacije o sinkronizaciji, JSON format podataka koji sadrži broj primljenih mjerenja 'received', broj novih mjerenja 'new_data' i broj zapisa mjerenja spremljenih u bazu podataka 'synced'. Svakom sinkronizacijom osvježavaju se podaci o glukometru u bazi podataka kroz model GlucometerDevice te i liječnik i pacijent u vlastitim sučeljima mogu vidjeti osvježene podatke.

3.3.4. Web aplikacija

Web aplikacija je jedan od dijelova cjelokupnog sustava koja liječnicima pruža napredne mogućnosti pregleda bitnih mjerenih vrijednosti od strane pacijenata uz mnogobrojne analize podataka i vizualizacije kroz raznorazne grafikone te funkcije upravljanja korisničkim računima pacijenata i ciljnim vrijednostima. Pacijentima se omogućuje pristup i pregled vlastitih mjerenja kao i pristup različitim statističkim analizama. Pored ovih mogućnosti, nude se i mogućnosti izvoda podataka na različite načine u različitim formatima. U nastavku je u svakom potpoglavlju objašnjen po jedan dio funkcionalnosti potkrijepljen dodatnim informacijama o načinu na koji je razvijen.

3.3.4.1. Prikaz pacijenata

Početni prikaz dijela web aplikacije koju koristi liječnik je popis pacijenata. Ovdje su na pregledan način ispisani bitni podaci pacijenata te korisnik dobiva brz pregled stanja pacijenata kroz zadnjih tjedan dana.

E-Gluko

Dr Doktor

Online

Izbornik

Pacijenti

Novi pacijent

Pacijenti

popis svih pacijenata

	Pacijent	Datum rođenja	Dijabetes od	Zadnje mjerenje	AVG vrijednosti zadnjih 7 dana	
					Kretanje (dn.) Kalorije (dn.)	GUK Bolus/bazalni (dn.)
Prikaži	Luka Bartolić DT2	20.05.1993. (24 god.)	15.02.2013. (4 god.)	Srijeda, 24.01.2018. 23:59 Koraci	1889 koraci 278 kcal MJERENO DANA: 4	11.2 mmol/l UK. OČITAVANJA: 17 17 U / 17 U UK. DOZIRANJA: 19 / 7
Prikaži	Acme Emca DT2	20.05.1993. (24 god.)	15.02.2013. (4 god.)	Četvrtak, 25.01.2018. 09:14 Razina glukoze	nema mjerenja	6.5 mmol/l UK. OČITAVANJA: 17 17 U / 16 U UK. DOZIRANJA: 19 / 6
Prikaži	Test ime Test prezime DT1	21.06.1995. (22 god.)	20.01.2012. (6 god.)	nema mjerenja	nema mjerenja	nema mjerenja

Slika 3.9. Prikaz popisa pacijenata za liječnika.

Osim osnovnih podataka poput imena i prezimena te tipa dijabetesa od kojeg pacijent boluje, za svakog pacijenta prikazan je datum i vrijeme zadnjeg mjerenja te vrsta zapisa mjerenja. Međutim, najbitnija značajka je statistički pregled stanja u zadnjih tjedan dana na temelju prikupljenih i obrađenih podataka. Prikazani su statistički podaci prosječnog dnevnog kretanja i prosječne dnevne potrošnje kalorija kako bi se dobio generalni uvid o aktivnosti pacijenta. Pored toga, korisniku je izračunata i statistika razine glukoze u krvi te količina i odnos unosa doza inzulina. Izračunate vrijednosti popraćene su indikatorima koji označavaju odstupanja od ciljnih vrijednosti ili kritična stanja na temelju ciljnih vrijednosti za taj period.

3.3.4.2. Kreiranje pacijenta

Odabirom stavke „Novi pacijent“ u izborniku otvara se prikaz s obrascima za dodavanje pacijenta. Popunjavanjem obrasca i potvrdom za unosom pacijenta poziva se metoda *store()* u kontroleru *PatientController*, a kao prvi parametar prima zahtjev, odnosno instancu *StorePatientRequest* objekta. Validacija podataka, odnosno ispitivanje ispravnosti unesenih podataka vrši se na strani poslužitelja kroz *StorePatientRequest* klasu koja nasljeđuje *Laravelovu FormRequest* klasu. Laravel kroz *FormRequest* klasu nudi mogućnost autorizacije i validacije podataka na temelju postavljenih pravila i uvjeta. U metodi *rules()* unutar *StorePatientRequest* klase definirana su pravila, a u metodi *messages()* povratne poruke grešaka koje se vraćaju kao odgovor na zahtjev ukoliko validacija nije prošla. Ukoliko validacija ne prođe, korisniku se u korisničkom sučelju ispisuju poruke grešaka te nakon ispravljanja pacijent može biti dodan. Programski kod na strani poslužitelja zadužen za kreiranje novog pacijenta nalazi se u *PatientRepository* klasi te je ista metoda iskoristiva i za poziv prilikom operacije *update*, odnosno uređivanja podataka pacijenta.

Metode iz klase repozitorija pozivaju se iz kontrolera, naprimjer:

```
$user = $this->patientRepo->storeOrUpdate($request);
```

The screenshot shows the 'Pacijenti' form in the E-Gluko application. The form is titled 'Pacijenti kreiranje novog'. It has two main sections: 'Općeniti podaci' and 'Dodatni podaci'. The 'Općeniti podaci' section contains fields for 'Ime' (Name), 'Prezime' (Surname), 'Email', 'Lozinka' (Password), and 'Status' (Aktiviran/Neaktiviran). The 'Dodatni podaci' section contains fields for 'Datum rođenja' (Date of birth), 'Visina' (Height), 'Spol' (Gender), 'Dijabetes od' (Diabetes since), and 'Tip dijabetesa' (Type of diabetes). A 'Spremi' (Save) button is located at the bottom right of the form.

Slika 3.10. Obrasci za dodavanje novog pacijenta.

Nakon što sustav doda pacijenta u bazu podataka, korisnik (liječnik) mu mora postaviti prve ciljne vrijednosti. Ukoliko ciljne vrijednosti nisu postavljene, pristup pacijentu biti će onemogućen jer su za potrebe detaljne statističke obrade i analize potrebne definirane ciljne vrijednosti. U ovu svrhu koristi se koncept *middlewarea*. *Middleware* pruža prikladan mehanizam za filtriranje zahtjeva HTTP-a koji pristižu na stranu poslužitelja. U Laravel razvojnom okviru stvara se izradom klase u kojoj se implementira glavna metoda *handle(Request \$request, Closure \$next)*. Ta metoda kao parametre prima zahtjev *\$request* i povratnu metodu *\$next*. Unutar glavne metode *handle(...)* definira se zadatak koji se mora izvršiti prije propuštanja zahtjeva prema kontroleru. U ovom slučaju provjerava se ima li pacijent postavljen barem jedan skup ciljnih vrijednosti pa je metoda implementirana na sljedeći način:

```
public function handle(Request $request, Closure $next) {  
    $user = User::where('id', $request->route('patient'))->first();  
    $healthGoals = $user->healthGoals()->get();  
    if ($healthGoals->count() == 0) {  
        return redirect()->route('admin.health-goals.index', $user->id);  
    }  
    return $next($request);  
}
```

Ukoliko pacijent ima postavljene ciljne vrijednosti, *middleware* će propustiti zahtjev prema kontroleru koji će ga obraditi, a ako nema, korisnik će biti preusmjeren na prikaz web aplikacije za dodavanje ciljnih vrijednosti.

E-Gluko

Dr Doktor
Online

Izbornik

Pacijenti

Test ime Test prezime

Pregled

Trendovi

Zapisi i izvještaji

Statistika

Ciljne vrijednosti

Novi pacijent

Ciljne vrijednosti popis svih ciljnih vrijednosti

⚠ Ciljne vrijednosti nisu postavljene, potrebno ih je postaviti.

Dodavanje novih ciljnih vrijednosti

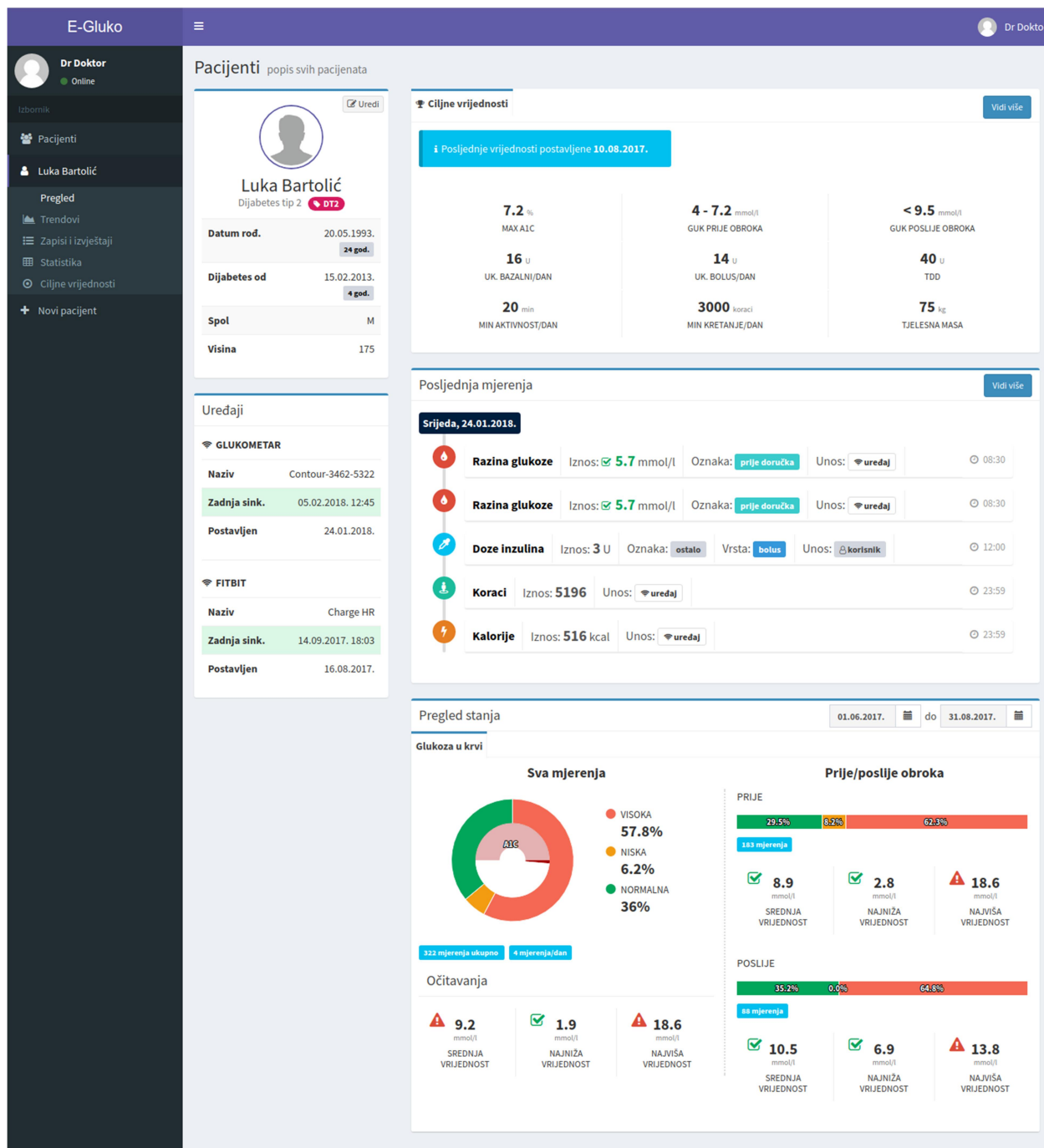
MAX A1C [%]	MIN prije obroka [mmol/l]	MAX prije obroka [mmol/l]	M/
Uk. bazalni/dan [U]	Uk. bolus/dan [U]	TDD [U]	
MIN aktivnost/dan [min]	MIN kretanje/dan [koraci]	Tjelesna masa [kg]	Dž

Slika 3.11. Onemogućenost pristupa pacijentu zbog nepostavljenih ciljnih vrijednosti.

Postoje različite primjene ovog koncepta, primjerice u aspektu razvoja API-ja, u slučaju nepropuštanja zahtjeva mogao bi se definirati odgovarajuć JSON odgovor s povratnom informacijom i statusom ili kodom pogreške.

3.3.4.3. Profil pacijenta

Odabirom jednog od pacijenata iz liste pacijenata otvara se prikaz profila tog pacijenta.



Slika 3.11. Profil pacijenta.

Ovdje korisnik ima uvid u najbitnije informacije o pacijentu:

- osnovne informacije (ime i prezime, tip dijabetesa, datum rođenja, datum dijagnoze dijabetesa, spol i visina)
- aktualne (posljednje postavljene) ciljne vrijednosti

- posljednja mjerenja (lista nedavnih mjerenja s dodatnim informacijama i indikatorima odstupanja od ciljnih vrijednosti)
 - ovaj dio sučelja profila koristi parcijalan prikaz (engl. *partial view*) izveden iz dijela sustava „Zapisi i izvještaji“ i daje primjer ponovne iskoristivosti funkcije sustava te je potpuna funkcionalnost opisana u poglavlju „Pregled zapisa i izvještaja“
- popis povezanih uređaja s pacijentom (uvid u stanje glukometra i Fitbit uređaja) gdje su prikazane informacije poput naziva uređaja i zadnje sinkronizacije podataka
 - informacije su osvježene u bazi podataka i korisničkom sučelju svaki put kad pacijentovi uređaji izvrše sinkronizaciju podataka
- pregled stanja razine glukoze u krvi

Pregled stanja razine glukoze u krvi

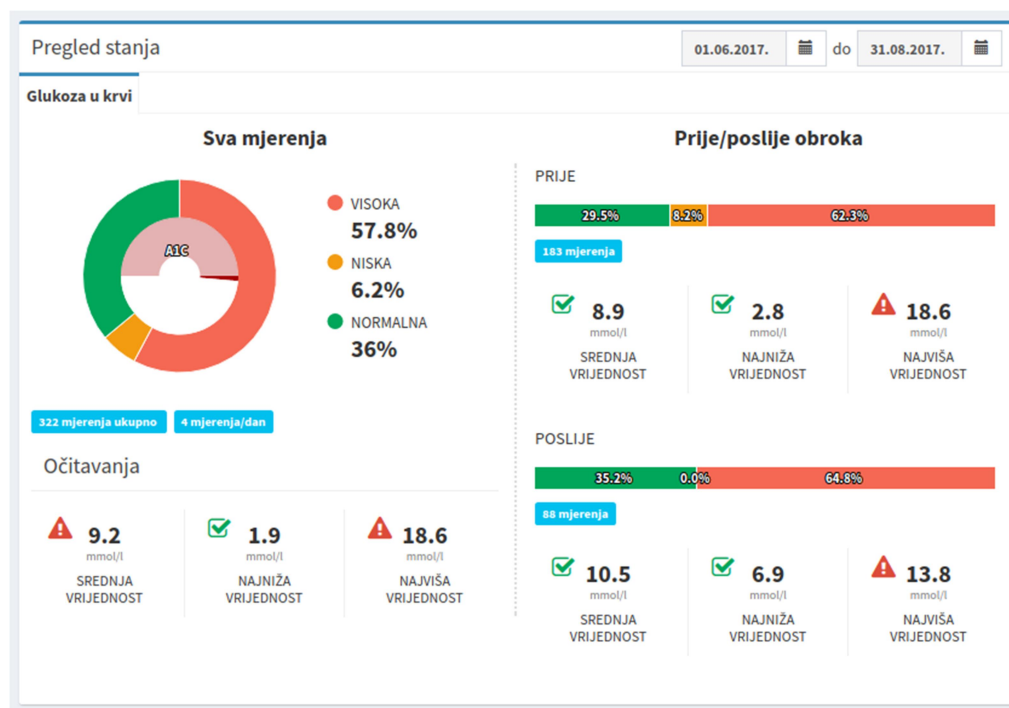
Ovaj dio sučelja korisniku pruža elegantan, interaktivan i dinamičan uvid u stanje razine glukoze u krvi za željeni period. Korisnik odabire raspon datuma (period), poslužiteljska programska podrška prikuplja podatke i obrađuje ih te se u sučelju prikazuju statistički podaci uz vizualizaciju. Sučelje sadrži:

- statistika i vizualizacija na temelju svih mjerenja u odabranom periodu
 - grafikon prikazuje zastupljenost visoke, niske i normalne razine glukoze u krvi s vrijednostima izraženima u postocima
 - unutar prstenastog grafikona prikazano je odstupanje estimirane A1C vrijednosti od ciljne vrijednosti za A1C izraženo u postocima uz podatak o točnoj vrijednosti
 - prikaz ukupnog broja mjerenja i prosječnog broja mjerenja na dan
 - srednja, najniža i najviša vrijednost očitavanja glukoze u krvi uz indikatore odstupanja
- odvojena statistika i vizualizacija na temelju oznaka mjerenja – prije i poslije obroka na sličan način kao i za sva mjerenja

Implementacija ovog dijela sučelja izvedena je na temelju različitih tehnologija koje su već prethodno opisane. S obzirom da je cilj bio na elegantan način pružiti najbitnije podatke o stanju pacijenta (stanje razine glukoze u krvi) uz vizualizaciju podataka, korišten je JavaScript programski jezik uz dodatak Handlebars *template enginea*, Highcharts biblioteka za vizualizaciju te AJAX tehnologija za asinkrono slanje zahtjeva prema serveru i preuzimanje skupa podataka. Funkcionalnost se izvodi na sljedeći način:

1. nakon renderiranja prikaza profila od strane poslužitelja poziva se JavaScript funkcija *setupGlucoseSummaryUI(true)*;
2. funkcija uzima predefiniran raspon datuma te poziva funkciju *getGlucoseSummary()* koja je odgovorna za izradu asinkronog zahtjeva prema poslužitelju
3. poslužitelj vraća očekivan skup podataka te se kroz *callback(data)* funkciju podaci vraćaju natrag funkciji *setupGlucoseSummaryUI()* koja zatim poziva funkciju Handlebars biblioteke *compile()* i predaje joj odgovarajuć predložak (engl. *template*) i podatke, zatim formatira podatke u odgovarajuć oblik za vizualizaciju i poziva funkcije za generiranje grafikona poput *drawGlucoseSummaryChart()*
4. nakon završetka prevođenja sučelja od strane Handlebarsa i generiranja grafikona od strane Highchartsa korisniku se prikazuju željeni podaci
5. korisnik ima mogućnost promjene raspona datuma te se navedena procedura izvodi ponovno na isti način

Na ovaj način omogućen je brz i jasan prikaz stanja pacijenta u odabranom periodu.



Slika 3.12. Pregled zdravstvenog stanja pacijenta s naglaskom na razinu glukoze.

3.3.4.4. Prikaz zdravstvenog stanja kroz trendove

Prikaz zdravstvenog stanja kroz trendove predstavlja kompleksniji dio web aplikacije, a obrada podataka zahtjevniji dio poslužitelja. Korisničko sučelje sastoji se od opcija vrste prikaza trendova, odabira vremenskog raspona i početnog datuma uz mogućnost pomaka ulijevo ili

udesno za odabran raspon (strelice lijevo-desno), analize prema vrsti zapisa prikazane kroz tablicu i interaktivnih grafikona s indikatorima odstupanja i dodatnim informacijama. U pozadini, na strani poslužitelja, izvršava se detaljna obrada podataka za odabranu vrstu prikaza te se sve vrijednosti uspoređuju s postavljenim ciljnim vrijednostima od strane liječnika za taj period. Krajnji korisnik ima mogućnost odabira prikaza trendova na dva načina: dnevni pregled kroz tjedan i vremenski pregled.

Dnevni pregled kroz tjedan omogućuje prikaz analize podataka i vizualizacije podataka za odabran vremenski raspon grupiranih po danima u tjednu.

Na ovaj način korisnik dobiva detaljan uvid u kretanje vrijednosti različitih mjerenja kroz dane u tjednu te se vrlo jasno može vidjeti i iščitati u kojim danima su mjerenja bila kritična ili u dozvoljenim rasponima vrijednosti. Analiza mjerenja prikazana u tablici prikazuje obrađene podatke kroz dane u tjednu na sljedeći način:

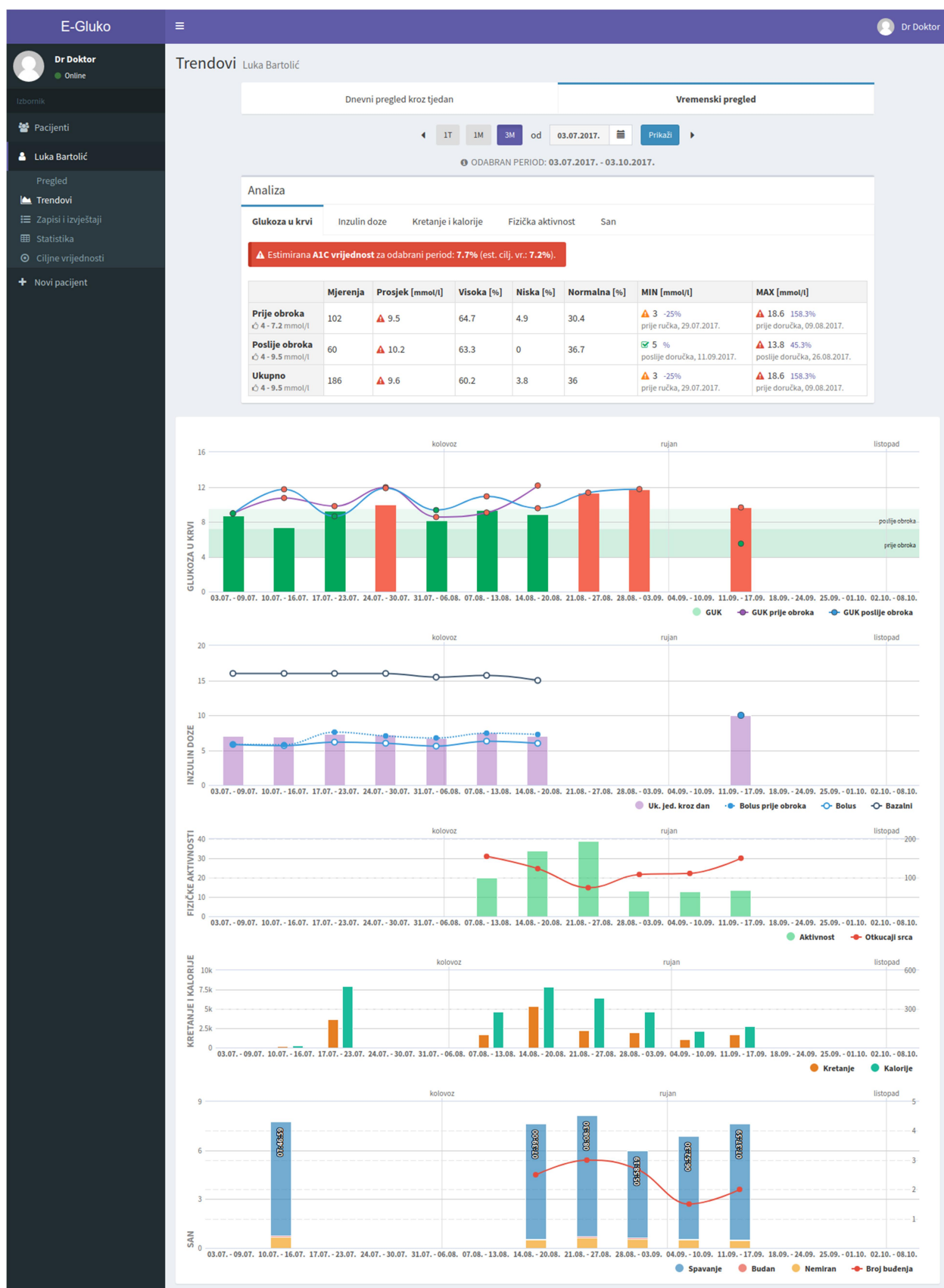
- za svaki dan u tjednu izračunate su prosječne vrijednosti, najniže vrijednosti i najviše vrijednosti
- indikatorima odstupanja prikazana su odstupanja od ciljnih vrijednosti, a odstupanje je dodatno izraženo u postocima pored vrijednosti mjerenja
- za najniže i najviše pronađene vrijednosti u odabranom periodu ispisan je datum tog mjerenja

Dodatno, za analizu glukoze u krvi za raspon datuma gdje je pronađen dovoljan broj mjerenja, estimirana je A1C vrijednost na temelju standardne formule za izračun A1C vrijednosti. S druge strane, kod analize doza inzulina izračunata je TDD vrijednost za odabran raspon datuma. Obje izračunate vrijednosti usporedive su s ciljnim vrijednostima te su u odnosu na ciljne vrijednosti na odgovarajuć način prikazane i vizualizirane.

Implementirani interaktivni grafikoni prikazuju sljedeća svojstva kroz dane u tjednu:

1. glukoza u krvi
 - kretanje srednje vrijednosti (dnevnog prosjeka)
 - kretanje srednje vrijednosti unutar pojedinog dana
 - konkretne vrijednosti mjerenja unutar pojedinog dana označenih odgovarajućom bojom u odnosu na ciljne vrijednosti
 - raspon ciljnih vrijednosti prije obroka i raspon poslije obroka
2. doze inzulina
 - kretanje srednje vrijednosti unutar pojedinog dana grupirano kroz sate odvojeno prema tipu inzulina
 - raspone kretanja vrijednosti (raspon između najniže i najviše vrijednosti doze inzulina u tom dijelu dana) unutar pojedinog dana također grupirano kroz sate odvojeno prema tipu inzulina
 - ukupni dnevni prosjek (oba tipa inzulina)
3. fizičke aktivnosti
 - ukupni dnevni prosjek trajanja aktivnosti
 - trajanje pojedine aktivnosti kroz sate
4. kretanje i kalorije
 - odvojeni ukupni dnevni prosjek kretanja i potrošnje kalorija
 - odvojeno kretanje najviših izmjerenih vrijednosti kretanja i potrošnje kalorija
5. san
 - na isti način kao i fizičke aktivnosti

Vremenski pregled, za razliku od dnevnog pregleda kroz tjedan, pruža analizu i vizualizaciju podataka kroz vrijeme za odabran period. Odabirom opcije prikaza raspona jednog tjedna (1T) ili jednog mjeseca (1M) obrađuju se podaci, generira analiza i prikazuje vizualizacija kontinuiranih vrijednosti mjerenja kroz taj period, a odabirom opcije prikaza od tri mjeseca (3M) vizualizacija je kroz grafikone prilagođena za prikaz podataka grupiranih po tjednima. Time je korisniku omogućeno pratiti mjerenja i stanje pacijenta kroz dane u odabranom periodu, ali i uspoređivati obrađene podatke kroz tjedne odabranog perioda. Analiza mjerenja u tabličnom prikazu prikazuje izračunate vrijednosti zajedno s indikatorima odstupanja u ovisnosti o ciljnim vrijednostima, a detaljnije je objašnjena u poglavlju „Statistika“,

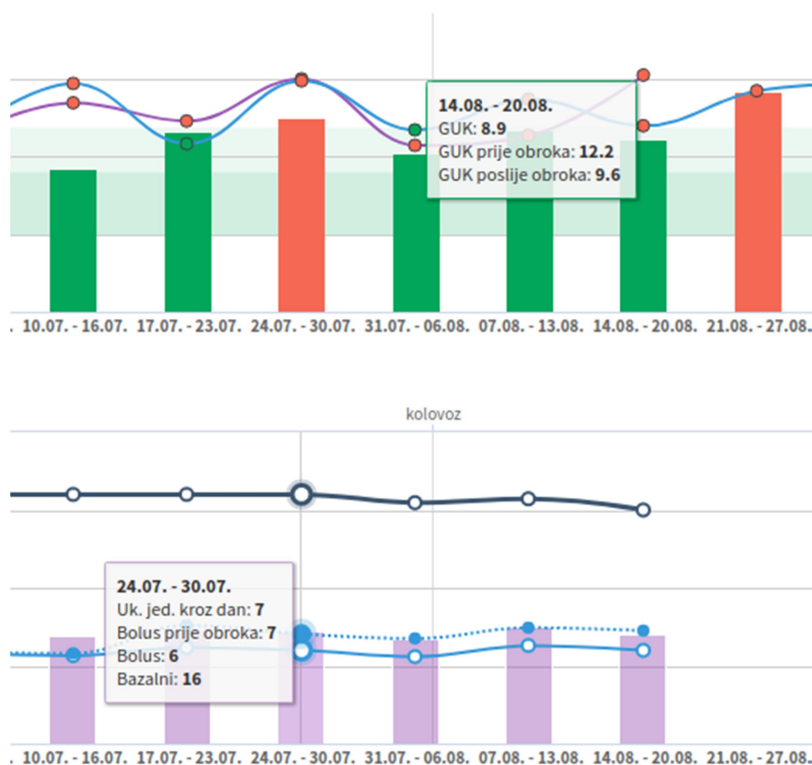


Slika 3.14. Trendovi – vremenski pregled (3M).

Grafikoni se za periode od 1T i 1M generiraju se na sličan način kao i kod dnevnog pregleda kroz tjedan, no vrijednosti nisu grupirane već su prikazana kretanja konkretnih izmjerenih vrijednosti kroz odabran vremenski raspon. S obzirom da se prikazuju točne vrijednosti mjerenja, u grafikonima su vizualizirana i kretanja dodatnih vrijednosti:

- glukoza u krvi: odvojene vrijednosti mjerenja prije i poslije obroka
- doze inzulina: zastupljenost ukupnih jedinica inzulina po danu
- fizičke aktivnosti: intenzitet rada srca (otkucaji srca)
- san: zastupljenost trajanja faze spavanja, budnog stanja i nemirnog stanja te kretanje vrijednosti broja buđenja i nemirnosti

Uz samu vizualizaciju, za svaku vrijednost na grafikonima implementirani su elementi koji prikazuju detaljniji opis vrijednosti, a pojavljuju se postavljanjem pokazivača na pojedinu vrijednost



Slika 3.15. Elementi koji sadrže dodatne informacije o pojedinoj vrijednosti u grafikonu.

3.3.4.5. Pregled zapisa i izvještaja

Pregled zapisa i izvještaja pruža korisniku kronološki pregled mjerenja grupiranih po danima te obrađenih i analiziranih statističkih podataka grupiranih za odabran tjedan i svaki dan zasebno. Sva mjerenja koja imaju definirane ciljne vrijednosti sadrže indikatore odstupanja, isto kao i tjedna i dnevna statistika. Na razini programskog koda, izlistavanje zapisa mjerenja napravljeno je na modularan način što omogućuje ponovno korištenje u različitim dijelovima web aplikacije. Primjer je ponovno korištenje dijela ovog korisničkog sučelja i logike renderiranja sučelja u prikazu koje predstavlja profil pacijenta. Korištenjem Laravelovog Blade *template enginea* napravljeni su predlošci koji primaju podatke kroz parametre te ih renderiraju u cjelokupno korisničko sučelje.

```
<ul class="timeline _logs-timeline">
    @foreach($logs['data'] as $dataKey => $dataValue)
        <li class="time-label">
            <span class="bg-navy">
                {{ \Carbon\Carbon::parse(
                    $dataValue['date'])->formatLocalized(config(
                        'application.time.locale_date_formats.default_day_date')) }}
            </span>
        </li>
        <li>
            <div class="timeline-item">
                <div class="box-body _stats-box">
                    @include('backend.layouts.health-logs.includes.total-stats',
                        [
                            'stats' => $dataValue['stats'],
                            'except' => ['activity_daily_avg']
                        ])
                </div>
            </div>
        </li>
        @include('backend.layouts.health-logs.includes.
            'log-list', [
                'logs' => $dataValue['logs']
            ])
    @endforeach
</ul>
```

Petlja *foreach* na početku iterira kroz podatke grupirane po danu dohvaćenih od strane poslužitelja, odnosno kontrolera zaduženog za vraćanje podataka u ovaj dio web aplikacije, te kroz *include* metode poziva predefinirane predloške koje popunjava proslijeđenim podacima kroz dane parametre. Lista zapisa mjerenja generira se predajom odredišta predloška metodi

include, što je u ovom slučaju *backend.layouts.health-logs.includes.log-list*, odnosno *log-list* predložak. Parametar *logs* koji je predan predlošku sadrži kolekciju objekata dohvaćenu iz polja *\$dataValue* pod ključem *logs*. Predložak *log-list* zatim iterira kroz kolekciju objekata tipa *HealthLog* i prema vrsti zapisa postavlja potrebne vrijednosti i primjenjuje određenu strukturu i stil.

Programski kod na strani poslužitelja zadužen za obradu i vraćanje obrađenih podataka koji omogućuje funkcionalnosti zapisa i izvještaja definiran je u kontroleru *HealthLogController* i repozitorij klasi *HealthLogRepository*. Kontroler najprije u metodi *showPatientLogs()* prihvaća zahtjev od strane klijenta i potrebne parametre te definira osnovne varijable poput objekata početnog i krajnjeg datuma, objekta korisnika za kojeg se dohvaćaju zapisi mjerenja te nakon tog poziva metodu iz klase repozitorija za dohvaćanje potrebnih podataka.

```
public function showPatientLogs(Request $request, $userId, $date = null) {
    if ($date == null) $date = $this->now;
    else $date = Carbon::parse($date);
    $dateFrom = $date->copy()->startOfWeek();
    $dateTo = $date->copy()->endOfWeek();
    $user = User::findOrFail($userId);
    $healthGoals = $this->healthGoalRepo->getPatientGoalsBetweenDates(
        $user, $dateFrom, $dateTo);
    $healthLogs = $this->healthLogRepo->getPatientLogsList(
        $user, $healthGoals, $dateFrom, $dateTo, true);
    return view('backend.layouts.health-logs.logs', [
        'logs' => $healthLogs,
        'user' => $user,
        'next' => $date->copy()->addWeeks(1)->toDateString(),
        'prev' => $date->copy()->subWeeks(1)->toDateString()
    ]);
}
```

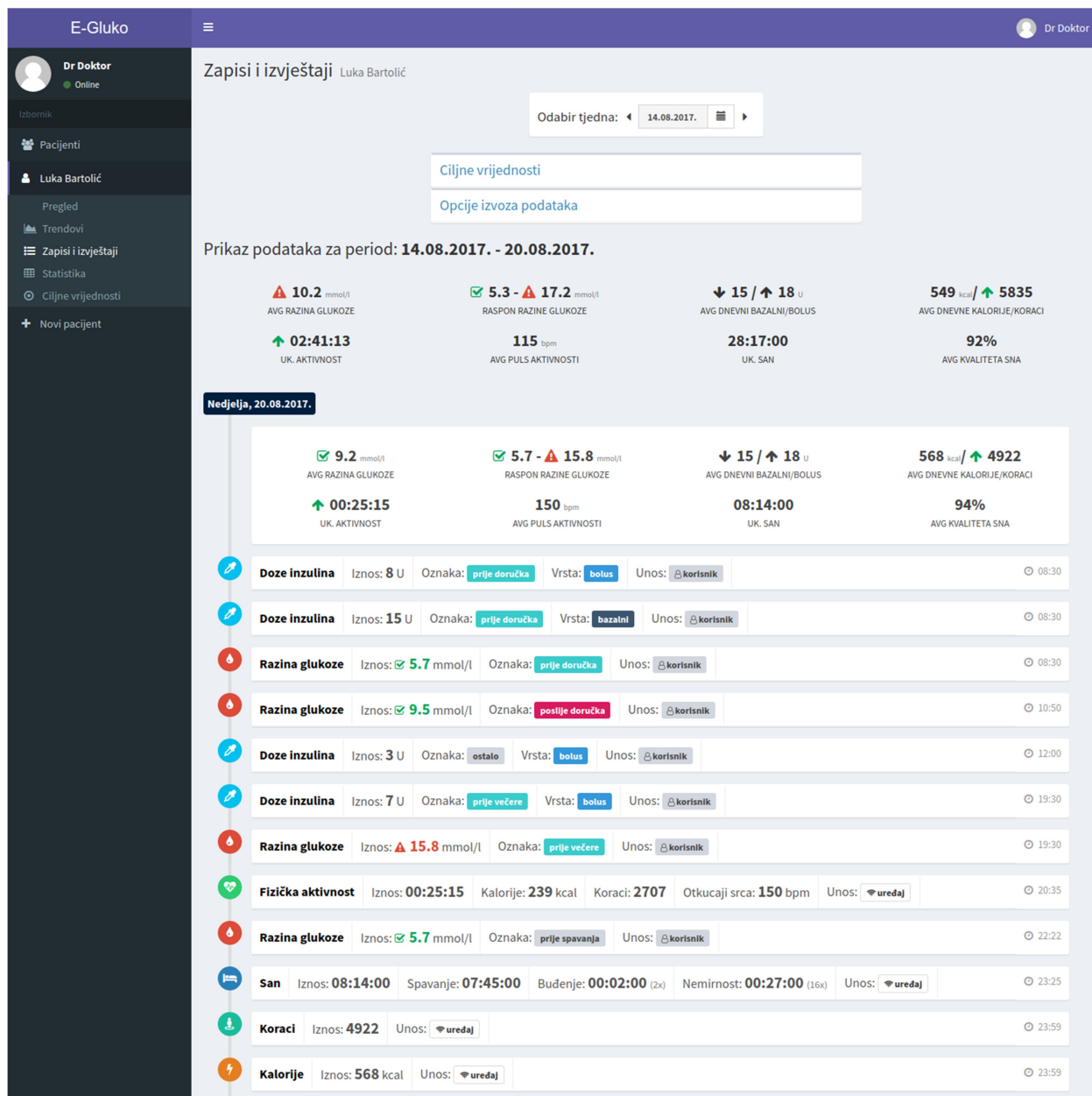
Metoda *getPatientLogsList()* klase *HealthLogRepository* zadužena je za definiranje logike dohvaćanja zapisa kako bi se kontroler rasteretio i kako bi se definirana programska logika mogla ponovno iskoristiti u drugim dijelovima programske podrške poslužitelja nevezanih uz navedenu metodu kontrolera. Metoda repozitorij klase objedinjuje dvije općenite metode koje su implementirane kako bi se mogle prema potrebi zajedno koristiti te ih poziva i kao rezultat kontroleru vraća rezultat koji sadrži strukturu podataka liste zapisa.

```

public function getPatientLogsList(User $user, Collection $healthGoals, $dateFrom,
$dateTo, $dailyStats = false, $sort = null) {
    $logs = $this->getPatientLogs($user, $dateFrom, $dateTo);
    $logsList = $this->createLogsList($logs, $healthGoals, $dateFrom, $dateTo,
        $dailyStats, $sort);

    return $logsList;
}

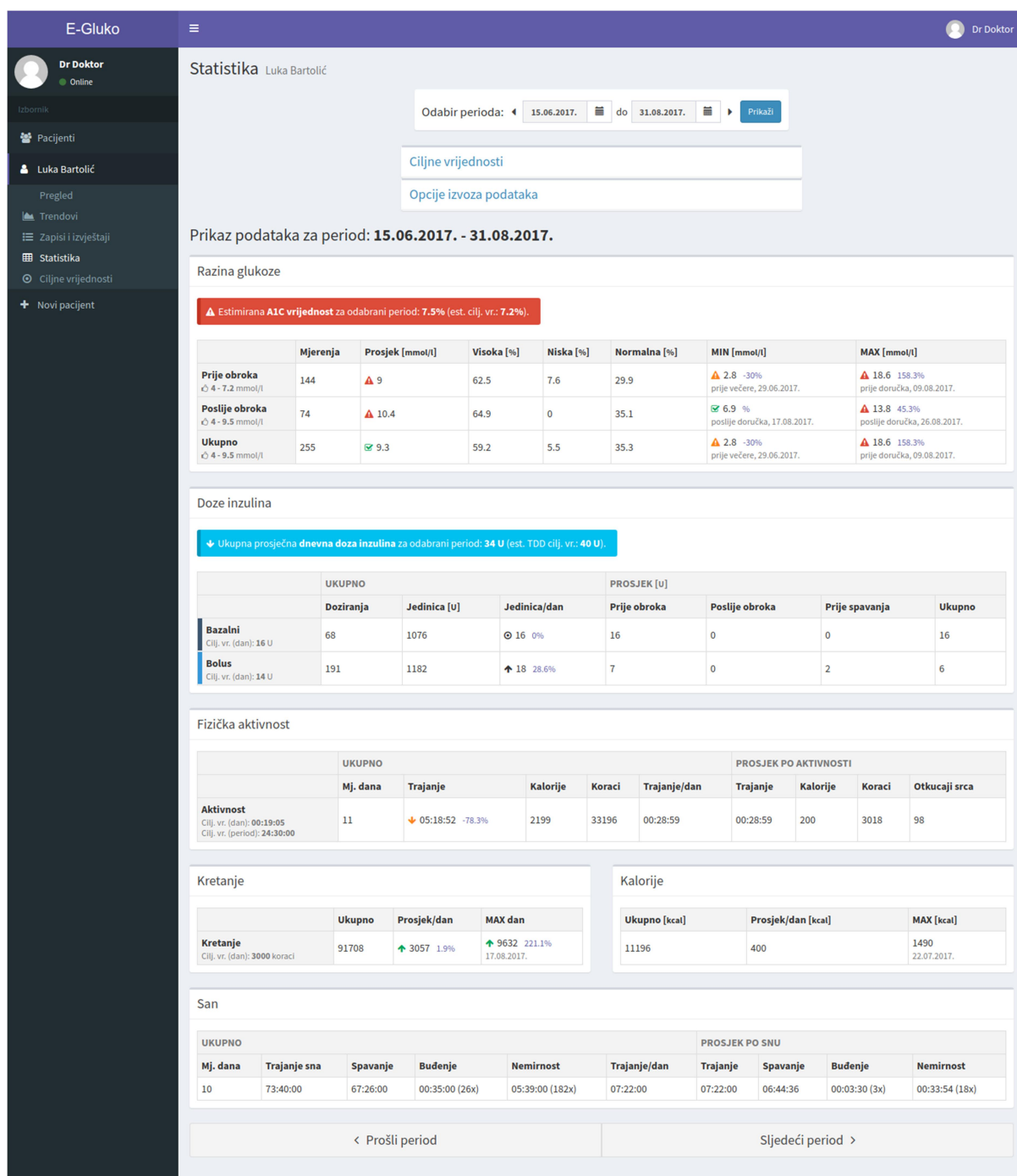
```



Slika 3.16. Prikaz zapisa i izvještaja.

3.3.4.6. Statistika

Pregled statistički obrađenih podataka za proizvoljan vremenski raspon omogućen je kroz ovo sučelje web aplikacije. Kroz ovaj dio web sučelja korisnik ima pristup detaljnoj analizi za odabran period, a sama implementacija koja uključuje programski kod kao i generiranje tabličnih prikaza napravljena je modularno te je time omogućena ponovna iskoristivost u sučelju web aplikacije „Trendovi“. Funkcionalnost je implementirana kroz metode kontrolera `HealthLogController`, repozitorij klase `HealthLogRepository` i dodatne metode iz klase repozitorija različitih vrsta zapisa mjerenja kao što su `GlucoseLogRepository` i `InsulinLogRepository`. Metoda kontrolera `showPatientStats()` prihvata zahtjev te poziva metodu `getSeparatedLogStats()` klase `HealthLogRepository`. S obzirom da svaka vrsta zapisa mjerenja ima implementiranu metodu koja je zadužena za statističku obradu i manipulaciju podataka, navedena metoda klase `HealthLogRepository` vraća odvojene statističke strukture obrađenih podataka prema vrsti zapisa. Primjer strukture podataka (polja) koja je rezultat statističke obrade vidljiva je na 3.17. gdje su prikazani obrađeni podaci za vrste zapisa *glucose* i *physical_activity*, odnosno razinu glukoze u krvi i fizičku aktivnost.

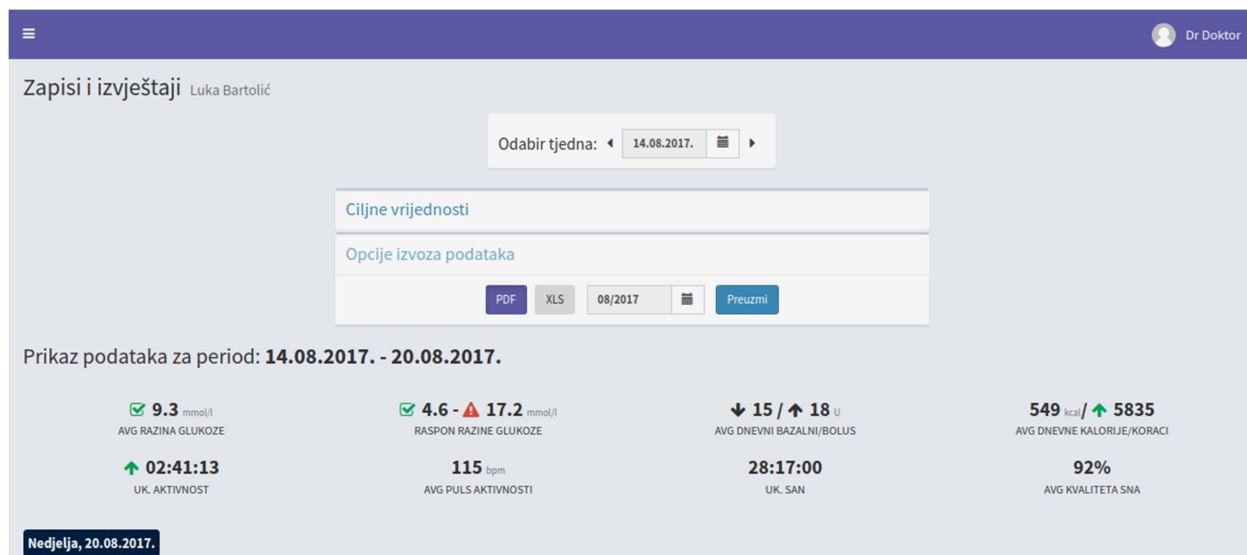


Slika 3.17. Statistika različitih vrsta mjerenja pacijenta.

3.3.4.7. Izvoz obrađenih i neobrađenih podataka

Izvoz obrađenih i neobrađenih podataka kroz sučelja web aplikacije „Zapisi i izvještaji“ te „Statistika“ pruža mogućnosti preuzimanja skupova podataka u PDF ili XLS formatu. Prilikom izvoza podataka koji se tiču zapisa i izvještaja, korisnik može odabrati u kojem formatu želi izvesti podatke. Odabirom opcije PDF programsko rješenje na strani poslužitelja će generirati PDF dokument koji će sadržavati prikaz zapisa mjerenja sa pridruženom statističkom obradom koristeći slučajnu logiku kao što koristi sučelje „Zapisi i izvještaji“ za izlistavanje zapisa

mjerenja. S druge strane, odabirom opcije XLS, korisniku se generiraju „sirovi“, odnosno izvorni podaci grupirani po stranicama XLS dokumenta prema vrsti zapisa mjerenja. S obzirom da se većina podataka prikuplja automatiziranim putem, liječnik primjerice može preuzeti svježije podatke u izvornom obliku te nad njima raditi dodatnu željenu obradu ukoliko funkcionalnosti sustava kroz korisničko sučelje ne pružaju potrebnu obradu.



Slika 3.18. Opcije izvoza podataka u prikazu „Zapisi i izvještaji“.

Za potrebe implementacije generiranja PDF i XLS dokumenata korišteni su paketi (biblioteke) *Laravel-DOMPDF* i *Laravel-Excel* koje pružaju elegantnu podršku Laravel razvojnom okviru postojećim bibliotekama za navedene primjene. *Laravel-DOMPDF* kao temelje koristi mogućnosti *DOMPDF* biblioteke, dok *Laravel-Excel* koristi mogućnosti *PHPExcel* biblioteke. Primjeri generiranih datoteka vidljivi su na 3.19 i 3.20. Primjer generiranja PDF dokumenta:

```
$pdf = PDF::loadView('pdf.health-logs.stats', [
    'stats' => $stats,
    'user' => $user,
    'dateFrom' => $dateFrom,
    'dateTo' => $dateTo
]);
return $pdf->download("{ $fileName }.pdf");
```

logs-luka-bartolic-2017-08-01-2017-08-31 (2).pdf									
7 / 20									
uredaj 23:59 Kalorije 386 kcal									
Petak, 18.08.2017.									
AVG razina glukoze [mmol/l]	Raspon razine glukoze [mmol/l]	AVG dnevni bazalni/bolus [U]	AVG dnevne kalorije/koraci [kcal/koraci]	Uk. aktivnosti	AVG puls aktivnosti [bpm]	Uk. san	AVG kvaliteta sna		
7.1	5.3 - 8.2	15 / 18	53 / 3294	00:17:56	79	00:00:00	-%		
		Unos	Iznos	Oznaka	Vrsta	Kalorije	Koraci	Otkucaji srca	Spavanje
korisnik	08:00	Doze inzulina	8 U	ostalo	bolus				
korisnik	08:00	Doze inzulina	15 U	ostalo	bazalni				
korisnik	10:46	Razina glukoze	8.2 mmol/l	poslije doručka					
korisnik	12:00	Doze inzulina	6 U	ostalo	bolus				
uredaj	16:24	Fizička aktivnost	00:17:56			70 kcal	1920	79 bpm	
korisnik	18:00	Doze inzulina	4 U	prije večere	bolus				
korisnik	18:00	Razina glukoze	5.3 mmol/l	prije večere					
korisnik	23:15	Razina glukoze	7.8 mmol/l	prije spavanja					
uredaj	23:59	Koraci	1936						
uredaj	23:59	Kalorije	53 kcal						
Četvrtak, 17.08.2017.									
AVG razina glukoze [mmol/l]	Raspon razine glukoze [mmol/l]	AVG dnevni bazalni/bolus [U]	AVG dnevne kalorije/koraci [kcal/koraci]	Uk. aktivnosti	AVG puls aktivnosti [bpm]	Uk. san	AVG kvaliteta sna		
7.9	6.9 - 8.8	15 / 14	1034 / 3294	00:45:22	97	05:39:00	91%		
		Unos	Iznos	Oznaka	Vrsta	Kalorije	Koraci	Otkucaji srca	Spavanje

Slika 3.19. Primjer generiranog PDF-a zapisa i izvještaja.

Primjer generiranja XLS dokumenta:

```
return Excel::create("{fileName}", function($excel) use($logsExport,
$sharedHeadings) {
    foreach($logsExport as $key => $value) {
        $excel->sheet($key, function($sheet) use($value, $key, $sharedHeadings) {
            $headingArr = array_merge($sharedHeadings,
                config('application.excel.headings.health_log.models.' . $key));
            $sheet->fromArray($value, null, 'A1', true, false);
            $sheet->prependRow($headingArr);
        });
    }
})->export('xls');
```

	A	B	C	D	E	F
	iznos	vrijeme	unos (0 - korisnik, 1 - uređaj)	vremenska oznaka	vremenska oznaka	
1						
2	5	2017-08-01 08:30:00	0	1	prije doručka	
3	8.7	2017-08-01 10:55:00	0	2	poslije doručka	
4	7.4	2017-08-01 17:30:00	0	5	prije večere	
5	5.8	2017-08-01 23:25:00	0	7	prije spavanja	
6	4.4	2017-08-02 08:00:00	0	1	prije doručka	
7	9.7	2017-08-02 10:44:00	0	2	poslije doručka	
8	8.4	2017-08-02 12:49:00	0	3	prije ručka	
9	4.3	2017-08-02 17:30:00	0	5	prije večere	
10	12.1	2017-08-02 22:38:00	0	7	prije spavanja	
11	13.6	2017-08-03 05:30:00	0	1	prije doručka	
12	8.7	2017-08-03 10:45:00	0	2	poslije doručka	
13	14.3	2017-08-03 18:00:00	0	5	prije večere	
14	13.7	2017-08-04 08:00:00	0	1	prije doručka	
15	7.5	2017-08-04 10:50:00	0	2	poslije doručka	
16	13	2017-08-05 08:00:00	0	1	prije doručka	
17	12.7	2017-08-05 10:42:00	0	2	poslije doručka	
18	9.5	2017-08-05 18:00:00	0	5	prije večere	
19	6.1	2017-08-05 22:30:00	0	7	prije spavanja	
20	5.6	2017-08-06 08:22:00	0	1	prije doručka	
21	8.2	2017-08-06 10:45:00	0	2	poslije doručka	
22	8.7	2017-08-06 18:15:00	0	5	prije večere	
23	9.2	2017-08-07 07:51:00	0	1	prije doručka	

Slika 3.20. Primjer generirano XLS-a zapisa i izvještaja.

Pored ovih datoteka, korisnik može također izvesti i preuzeti statistički obrađene podatke za željeni vremenski raspon kroz sučelje „Statistika“ u PDF formatu.

3.3.4.8. Upravljanje ciljnim vrijednostima

Krajnji korisnik, u ovom slučaju isključivo liječnik, kroz korisničko sučelje definirano u dijelu web aplikacije „Ciljne vrijednosti“ može upravljati ciljnim vrijednostima pacijenta. Liječniku je omogućen pregled promjena ciljnih vrijednosti koje u slučajevima kad je to nužno može obrisati, a također može i postaviti nove ciljne vrijednosti kroz obrazac smješten u istom prikazu aplikacije. Polja u obrascu za unos novih ciljnih vrijednosti su unaprijed popunjena zadnjim postavljenim ciljnim vrijednostima tako da je liječniku smanjen napor u definiranju novih te ih može brzo postaviti. Za uspješnu validaciju podataka koja se provodi na strani poslužitelja potrebno je unijeti sve podatke nakon čega su nove ciljne vrijednosti važeće od postavljenog datuma te se primjenjuju na sva mjerenja od tog datuma nadalje.

E-Gluko Dr Doktor Online

Ciljne vrijednosti popis svih ciljnih vrijednosti

Posljednje vrijednosti postavljene 10.08.2017.

GLUKOZA U KRVU (GUK)			INZULIN						
MAX A1C	Raspon prije obroka	MAX poslije obroka	Uk. bazalni/dan	Uk. bolus/dan	TDD	MIN aktivnost/dan	MIN kretanje/dan	Tjelesna masa	Datum od
7.5 %	4 - 7.5 mmol/l	10 mmol/l	14 U	15 U	32 U	15 min	2500 koraci	80 kg	01.03.2017. (sri)
7.2 %	4 - 7.2 mmol/l	9.5 mmol/l	16 U	14 U	40 U	15 min	3000 koraci	75 kg	13.05.2017. (sub)
7.2 %	4 - 7.2 mmol/l	9.5 mmol/l	16 U	14 U	40 U	20 min	3000 koraci	75 kg	10.08.2017. (čet)

Dodavanje novih ciljnih vrijednosti

MAX A1C [%]: 7.2 MIN prije obroka [mmol/l]: 4 MAX prije obroka [mmol/l]: 7.2 MAX poslije obroka [mmol/l]: 9.5

Uk. bazalni/dan [U]: 16 Uk. bolus/dan [U]: 14 TDD [U]: 40

MIN aktivnost/dan [min]: 20 MIN kretanje/dan [koraci]: 3000 Tjelesna masa [kg]: 75 Datum od: []

Spremi

Slika 3.21. Upravljanje ciljnim vrijednostima pacijenta.

3.3.5. Android mobilna aplikacija

Razvijena mobilna aplikacija orijentirana je prema pacijentima te im dozvoljava brz pristup najbitnijim informacijama i mjerenjima, pregled stanja i statistike te im omogućuje brz i jednostavan unos podataka te sinkronizaciju s vlastitim glukometrom. Mobilna aplikacija povezana je s poslužiteljskom uslugom (API-jem) te na temelju jedinstvenog API ključa putem HTTP protokola ostvaruje komunikaciju i pristupa resursima poslužitelja. Koristeći implementaciju BLE standarda, putem Bluetooth veze ostvaruje komunikaciju s glukometrom i omogućuje automatsku sinkronizaciju s podacima mjerenja u bazi podataka na strani poslužitelja.

3.3.5.1. Prijava

Prva Android aktivnost koja se učitava prilikom pokretanja aplikacije je MainActivity. Ova aktivnost jedino sadrži metodu koja provjerava jesu li korisnički podaci postavljeni, odnosno pohranjeni za buduće korištenje aplikacije:

```

public void callNextActivity() {
    Intent intent;
    if (User.isLoggedIn(this)) {
        intent = new Intent(this, LogsActivity.class);
    }
    else {
        intent = new Intent(this, LoginActivity.class);
    }
    startActivity(intent);
    finish();
}

```

Poziva se metoda na objektu User *isLoggedIn()* koja vraća *boolean* tip vrijednosti. Ukoliko je API ključ vrć postavljen, pokrenut će se aktivnost LogsActivity, a ukoliko nije, pokrenut će se aktivnost zadužena za prijavu LoginActivity.

LoginActivity je jednostavna aktivnost koja omogućuje pacijentu prijavu u sustav kroz mobilnu aplikaciju. U pozadini se nakon uspješne prijave prima API ključ koji se lokalno sprema za buduće korištenje usluga poslužitelja putem REST API-ja. Slanje zahtjeva za autentikacijom obavljeno je koristeći Retrofit biblioteku te je definirana metoda *postAuth()* koja prima parametre *email* i *password*. Kroz anotaciju *@POST* već je definirano da se radi o POST zahtjevu, a kao parametar je predan URI, odnosno *endpoint* ili adresa odredišta kojoj se pristupa. Nakon uspješne prijave pacijentu se prikazuje početno sučelje, odnosno LogsActivity.



Slika 3.22. Aktivnost za prijavu u mobilnu aplikaciju.

3.3.5.2. Tjedni zapisi

Tjedni zapisi predstavljaju polaznu točku nakon autentikacije u Android mobilnoj aplikaciji. Ovdje su pacijentu ispisani obrađeni podaci u obliku liste kroz dane odabranog tjedna. Ovdje su iskorištene komponente poput objekata ViewPager, TabLayout, Fragment, PagerAdapter, itd. Inicijalno učitavanje podataka aplikacija vrši iz LogsActivity aktivnosti pozivom metode *getLogsList()* iz HealthLogRepository klase koja definira potrebnu logiku prije poziva same HTTP operacije poput definiranja povratne (callback) metode uz definiranje povratnog tipa. Šalje se GET zahtjev web API-ju sustava, kao što je to definirano u HealthLogInterface Retrofit sučelju. Nakon primanja odgovora od strane poslužitelja, podaci se iz JSON formata pomoću Gson biblioteke konvertiraju u objekte definirane klase (HealthLogList). U svojstvu *listData* HealthLogList objekta nalaze se objekti tipa ListData koji zapravo predstavljaju zapise mjerenja grupiranih po danu. Na temelju podataka stvara se korisničko sučelje na dinamičan način pa se tako kroz LogsListDayPagerAdapter stvaraju tabovi koji predstavljaju dane a kroz fragmente se učitavaju zapisi mjerenja odabranog dana:

```
@Override
public CharSequence getPageTitle(int position) {
    return DateUtil.formatDateStrings(listData.get(position).getDate(),
        DateUtil.DB_DATE_FORMAT, DateUtil.APP_DATE_FORMAT_DAY).toUpperCase();
}
```

Uz pomoć biblioteke MaterialDateTimePicker pacijentu je omogućen jednostavan odabir tjedna za koji želi dohvatiti obrađene podatke. Brza navigacija prema prethodnom ili sljedećem tjednu omogućena je kroz gumbe (objekt Button) sa strelicama lijevo i desno. Odmah ispod izbornika tjedna prikazani su statistički podaci za cijeli period. Funkcionalnost je slična kao i u web aplikaciji – prikazana je vrijednost zajedno s indikatorom odstupanja i postotkom odstupanja od ciljne vrijednosti za ona mjerenja koja sadrže definirane ciljne vrijednosti.

Korisna značajka ove aktivnosti je vizualizacija kretanja razine glukoze u krvi kroz tjedan dana. Uz statističke podatke pacijent može dobiti brz uvid u kretanje vrijednosti očitavanja koja ne treba ni unositi budući da se unose bežično povezivanjem s glukometrom. U grafikonu je prikazana srednja vrijednost po danu uz prikaz točnih vrijednosti mjerenja gdje boje služe kao indikator odstupanja. Za bolje razumijevanje grafa iscrtane su ciljne vrijednosti koje vrijede za taj tjedan – najviša razina glukoze prije obroka i najviše razina glukoze poslije obroka koje bi pacijenti trebali nastojati ne prijeći. Grafikon je implementiran koristeći MPAndroidChart biblioteku.

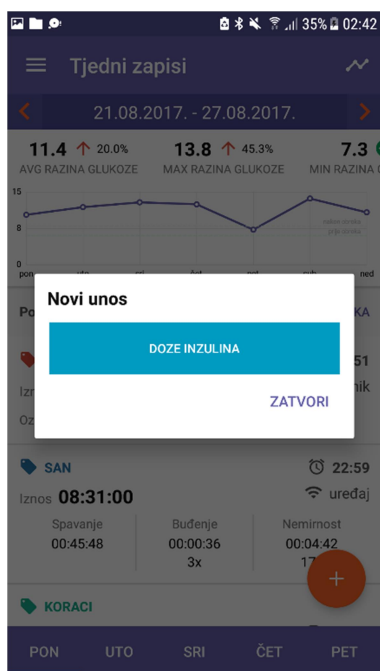
U cjelokupnom sustavu koriste se različita mjerenja i prikupljaju se različiti podaci, međutim postoji jedna vrsta zapisa mjerenja koja se mora unositi ručno, a to su doze inzulina.

Doze inzulina pacijent unosi kroz mobilnu aplikaciju gdje dodirom gumba tipa `FloatingActionButton` otvara izbornik i pristupa obrascu za brz unos doza inzulina. Najprije se radi GET HTTP zahtjev za dohvaćanjem bitnih podataka za prikaz u obrascu poput oznake mjerenja i tipa inzulina.

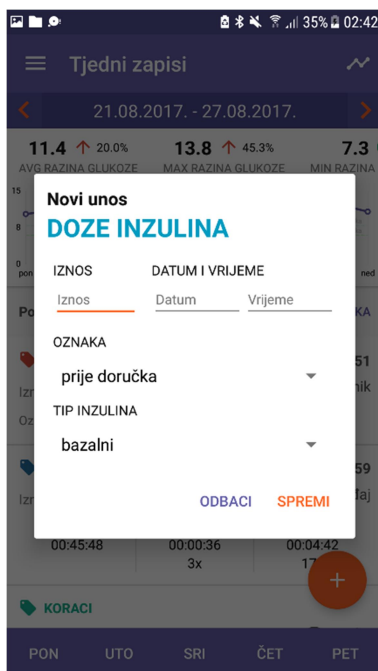
```
@GET("{api_key}/health-logs/create-data")  
Call<HealthLogCreateData> createData(@Path("api_key") String apiKey);
```

Implementirana je i validacija koristeći Android Saripaar biblioteku tako da pacijent ne može zabunom potvrditi unos praznih vrijednosti. Nakon uspješnog unosa podataka radi se POST HTTP zahtjev koji sadrži unesene podatke prema web API-ju.

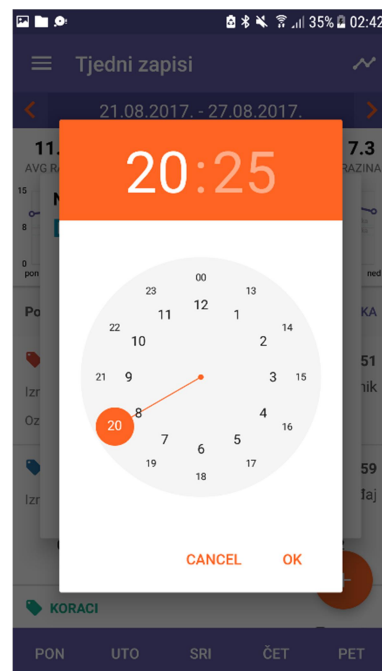
S obzirom da se u što većoj mjeri pokušava izbjeći ručni unos podataka i većina podataka se prikuplja automatiziranim putem, nije moguće brisati podatke, osim doza inzulina. Tu vrstu zapisa mjerenja moguće je obrisati dodirom na stavku u listi i odabirom opcije „Izbriši“.



Slika 3.23.a. Odabir dodavanja doza inzulina.



Slika 3.23.b. Dodavanje doza inzulina.

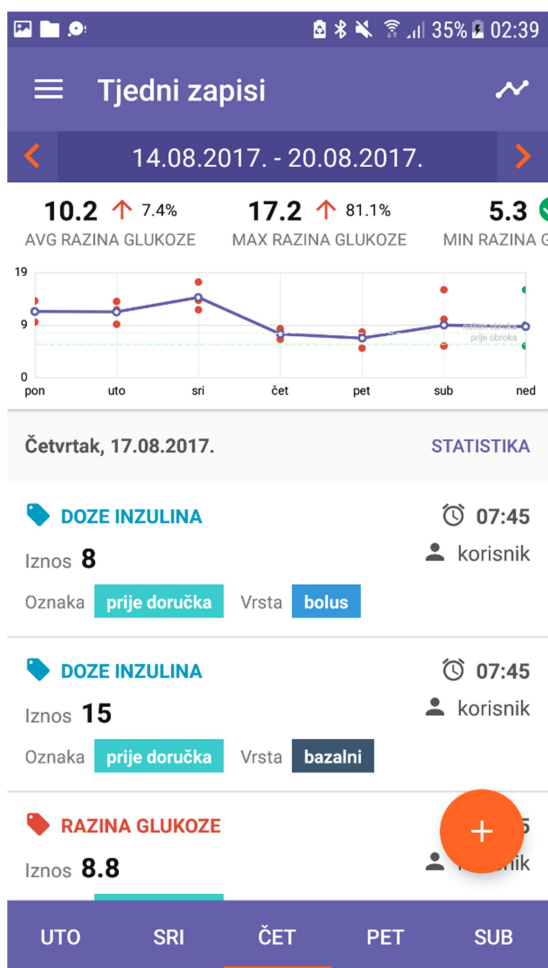


Slika 3.23.c. Definiranje vremena doze inzulina.

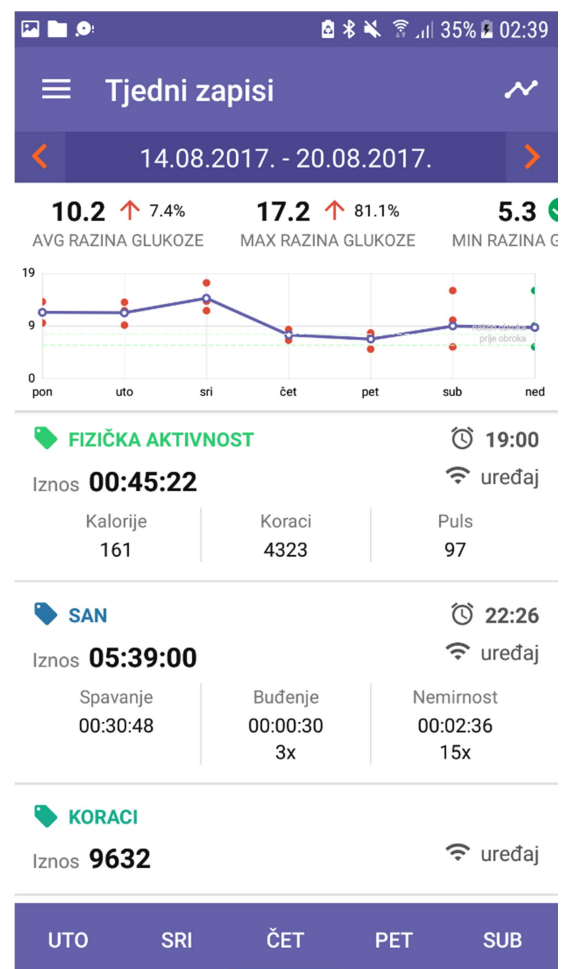
```

@FormUrlEncoded
@POST("{api_key}/health-logs")
Call<HealthLog> store(@Path("api_key") String apiKey,
                    @Field("value") Float value,
                    @Field("log_ts") String logTs,
                    @Field("log_type_key") String logTypeKey,
                    @Field("health_log_label_id") Integer labelId,
                    @Field("insulin_type_id") Integer insulinTypeId);

```



Slika 3.24.a. Tjedni zapisi koji prikazuju podatke obrađene na tjednoj bazi.

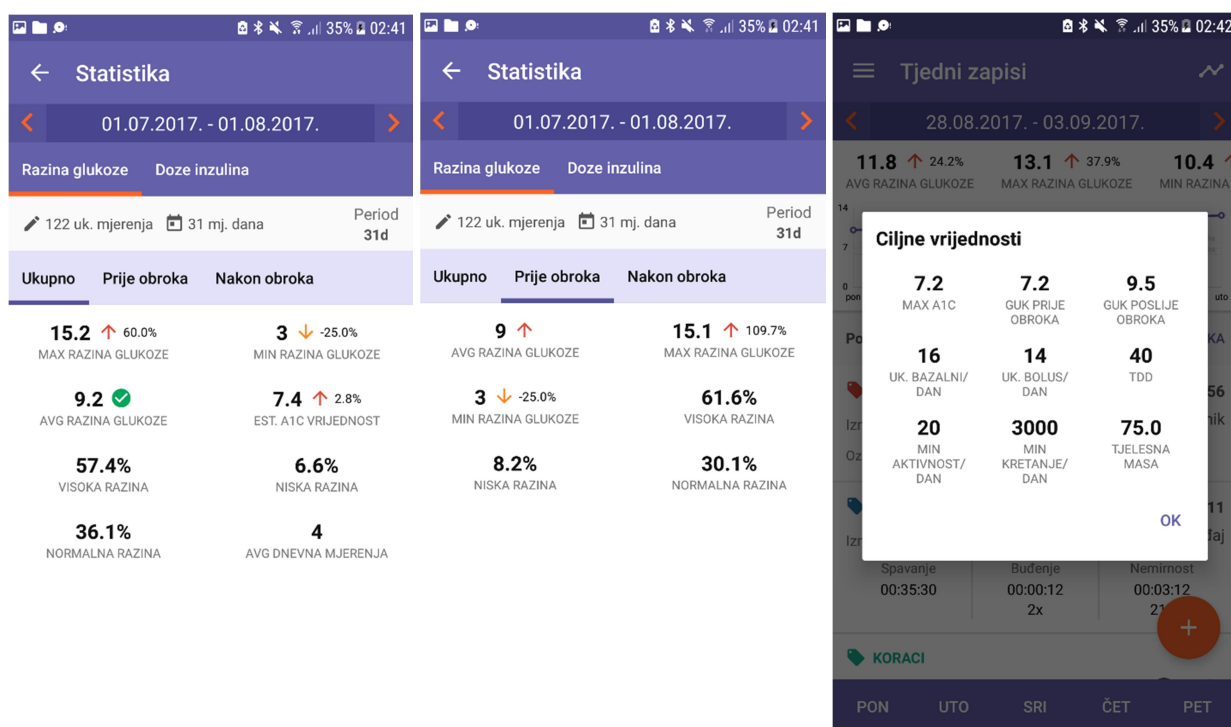


Slika 3.24.b. Tjedni zapisi koji prikazuju podatke obrađene na tjednoj bazi.

3.3.5.3. Statistika i ciljne vrijednosti

Pacijent koristeći mobilnu aplikaciju može dobiti brzi pregled statističkih podataka koji su vezani uz razinu glukoze i doze inzulina. Za obje vrste zapisa mjerenja moguće je vidjeti statistiku pa različitim kriterijima – za razinu glukoze ukupnu statistiku, prije obroka ili poslije obroka, a za doze inzuline ukupnu statistiku ili statistiku bazalnih ili bolus doza. Za vrijednosti

gdje je to moguće, korišteni su indikatori za prikaz kritičnih vrijednosti ili odstupanja od ciljnih vrijednosti. Statistika je prikazana za period od mjesec dana, generalne informacije o obrađenim podacima (broj ukupnog broja mjerenja, broj mjerenih dana i ukupan period) također su pružene pacijentu. Ova aktivnost implementirana je koristeći TotalStatsTypeFragment fragment, TotalStatsPagerAdapter te model podataka TotalStats koji kroz svoja svojstva uključuje StatsByType objekt i listu HealthGoal objekata. TotalStats je generalni model napravljen za primanje ukupne statistike prema nekoj vrsti mjerenja, a svaka podržana vrsta mjerenja u mobilnoj aplikaciji ima svoj model za parsiranje statističkih podataka.



Slika 3.25.a. Prikaz ukupne statistike razine glukoze.

Slika 3.25.b. Prikaz statistike razine glukoze prije obroka.

Slika 3.25.c. Ciljne vrijednosti pacijenta.

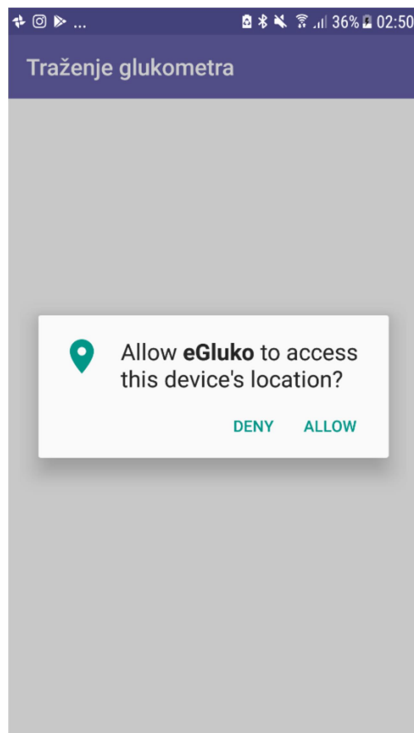
3.3.5.4. Sinkronizacija s glukometrom

Za potrebe sinkronizacije s Contour Next One glukometrom razvijena je programska podrška kroz Android aplikaciju. Programska podrška inicijalno je razvijena u odvojenom projektu te funkcionalnost povezivanja s glukometrom na temelju BLE standarda razvijeno prateći pravila standarda na vrlo elegantan način. Za potrebe povezivanja, održavanja veze i razmjenu podataka na temelju usluga i karakteristika GATT profila implementirana je klasa BleService, odnosno usluga, koja nasljeđuje Service klasu. Pored BleService klase kreirana je i klasa GlucoseRecord koja predstavlja model podataka kojeg će glukometar slati na zahtjev

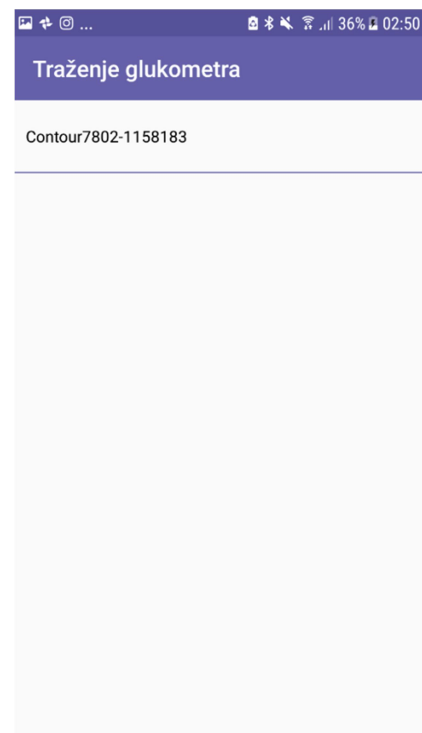
karakteristike. Metode `BleService` klase koja je razvijena odvojeno kako bi samostalno pružala usluge aktivnostima, aktivnosti pozivaju iz vlastitog konteksta.

Prilikom pokretanja aktivnosti „Glukometar“ pokreće se `GlucometerActivity` i provjerava se ima li korisnik aplikacije (pacijent) već podešen i postavljen glukometar (1). Ukoliko nema, automatski se pokreće skeniranje BLE uređaja (2). Na pojavu uređaja, pacijent ga odabire, podaci o uređaju spremaju se lokalno i na strani poslužitelja putem API-ja (3). Mobilna aplikacija se automatski nastoji povezati s glukometrom (4). Nakon ostvarivanja veze odmah kreće sa sinkronizacijom podataka (5). Definira se BLE usluga koja se koristi te se prvo putem određene karakteristike šalje zahtjev za obavještanjem o broju dostupnih podataka na uređaju (6). Ako postoje mjerenja na glukometru pacijenta, šalje se vrijednost druge karakteristike glukometru koja naznačuje da kao rezultat treba vratiti zapise glukoze u krvi (7). Nakon tog šalje se još jedan zahtjev putem pisanja karakteristike za dohvaćanje dodatnih podataka o mjerenjima kao što je odnos sa obrokom (prije/poslije obroka). Podaci se privremeno spremaju u listu (`List<GlucoseRecord>`) te kada su spremni, aktivnost koja koristi uslugu (servis) može pristupiti listi objekata `GlucoseRecord`. Kada aktivnost pristupi podacima, sinkronizira ih s onima na strani poslužitelja te pacijentu kroz aplikaciju vraća informacije o stanju sinkronizacije (8). Ovaj pojednostavljen tok u programskom kodu može se objasniti i na sljedeći način:

1. u `onCreate()` metodi poziva se metoda `isDeviceSetUp()`
 - ukoliko vrati neistinitu vrijednost, poziva se nova aktivnost `ScanBleActivity` te se nastavlja na korak 2.
 - ukoliko vrati istinitu vrijednost, nastavlja se na korak
2. u `ScanBleActivity` aktivnosti poziva se metoda `scanLeDevice()` koja sadrži logiku za skeniranje BLE uređaja te se pokreće skeniranje
 - implementirana je povratna funkcija koja na detekciju BLE uređaja uključuje uređaj u listu kako bi ga korisnik mogao odabrati



Slika 3.26. Dodjeljivanje prava za pristup lokacijskim uslugama što neki operacijski sustavi zahtjevaju prilikom korištenja BLE.

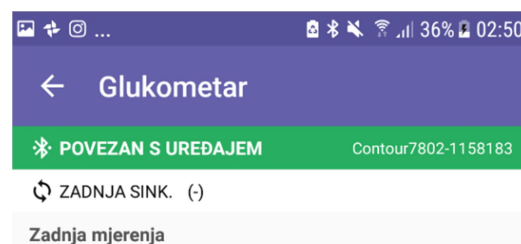
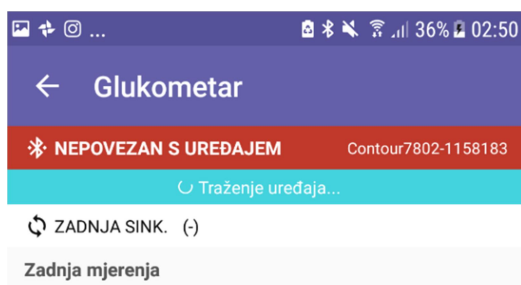


Slika 3.27. Rezultati skeniranja i pretraživanja glukometra (BLE uređaja).

```
private BluetoothAdapter.LeScanCallback bleScanCallback =
    new BluetoothAdapter.LeScanCallback() {
        @Override
        public void onLeScan(final BluetoothDevice device, int rssi, byte[]
scanRecord) {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    Log.d("DEVICE", device.getAddress() + " " + device.getName());
                    boolean deviceExists = false;
                    for(BluetoothDevice listDevice : btDevices) {
                        if (listDevice.getAddress().equals(device.getAddress())) {
                            deviceExists = true;
                        }
                    }
                    if (!deviceExists && device.getName() != null) {
                        btDevices.add(device);
                        btDevicesAdapter.notifyDataSetChanged();
                    }
                }
            });
        }
    }
```

- odabirom uređaja s popisa aktivnost poziva staru uz prosljeđivanje rezultata (naziva i adrese uređaja) kroz objekt Intent

3. poziva se *postStoreDevice()* metoda *GlucometerRepository* klase koja je zadužena za slanje zahtjeva za spremanjem uređaja pacijenta u bazu podataka kroz web API
4. poziva se metoda *setUpService()* koja najprije provjerava jesu li sva dopuštenja potrebnim uslugama dozvoljena te nakon toga povezuje uslugu s aktivnosti kroz Androidovu *bindService()* metodu i registrira *receivera* koji omogućuje primanje povratnih informacija u vezi komunikacije za vrijeme trajanja povezanosti
5. odmah nakon povezivanja s *BleService* uslugom, započinje korištenje njenih funkcija, a prva je *bleService.connect(deviceAddress)* – povezivanje s BLE uređajem na temelju definirane adrese



Uređaj uspješno postavljen

Čekanje na sinkronizaciju

Slika 3.28. Uspješno dodavanje uređaja, početak inicijalne konekcija s novdodanim uređajem.

Čekanje na sinkronizaciju

Slika 3.29. Uspješna povezanost s glukometrom, početak dohvaćanja podataka i sinkronizacije.

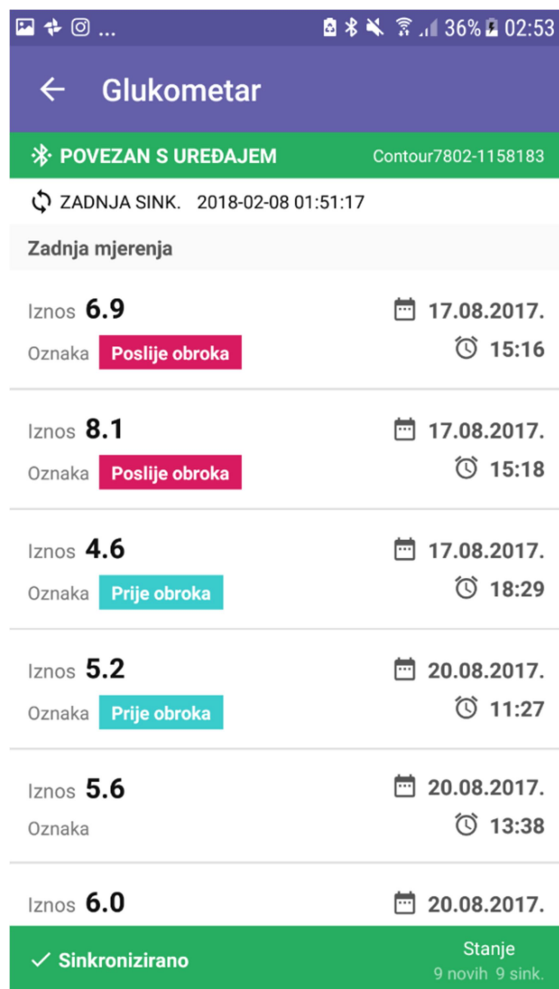
6. slanje karakteristike koja naznačuje potrebu za vraćanjem broja spremljenih podataka:

```
racpCharacteristic.setValue(new byte[2]);
racpCharacteristic.setValue(4, BluetoothGattCharacteristic.FORMAT_UINT8, 0);
racpCharacteristic.setValue(1, BluetoothGattCharacteristic.FORMAT_UINT8, 1);
```

7. ako postoje podaci, ponovno se šalje ta karakteristika ali s drugim vrijednostima koje naznačuju zahtjev za dohvaćanje podataka s uređaja
8. za dohvaćane podatke vrši se pretvorba u odgovarajuć tip podatka za pojedine dijelove vraćene vrijednosti na temelju specifikacije profila i karakteristika, poput dohvaćanja konteksta mjerenja koji predstavlja opcionalne parametre mjerenja (npr. odnos sa obrokom – prije ili poslije):

```
GlucoseRecord.MeasurementContext measurementContext = new GlucoseRecord.MeasurementContext();
flags = characteristic.getIntValue(BluetoothGattCharacteristic.FORMAT_UINT8, offset);
offset += 1;
boolean moreFlagsPresent = (flags & 0x80) > 0;
boolean carbohydratePresent = (flags & 0x01) > 0;
boolean mealPresent = (flags & 0x02) > 0;
measurementContext.sequenceNumber = characteristic.getIntValue(
    BluetoothGattCharacteristic.FORMAT_UINT16, offset);
offset += 2;
if (moreFlagsPresent) offset += 1;
if (carbohydratePresent) offset += 3;
if (mealPresent) {
    measurementContext.meal = characteristic.getIntValue(
        BluetoothGattCharacteristic.FORMAT_UINT8, offset);
}
GlucoseRecord record = glucoseRecords.get(measurementContext.sequenceNumber);
record.context = measurementContext;
```

- nakon toga aktivnost GlucometerActivity reagira na događaj koji naznačuje uspješno dohvaćanje podataka (*ACTION_RACP_RESPONSE_SUCCESS*) te šalje polje podataka tipa GlucoseRecord u JSON formatu prema poslužitelju



Slika 3.30. Poruka o uspješnoj sinkronizaciji, prikaz zapisa koji su pohranjeni na glukometru, informacija od strane poslužitelja o stanju sinkronizacije (dolje desno).

4. ZAKLJUČAK

Razvijeni sustav namjenjen praćenju zdravstvenog stanja oboljelih od dijabetesa demonstrira primjenu suvremenih informacijskih i komunikacijskih tehnologija u zdravstvu. Kroz automatizirano prikupljanje podataka, temeljitu obradu, statističku analizu i vizualizaciju podataka olakšava pacijentu i liječniku praćenje zdravstvenog stanja u odnosu na ciljne vrijednosti u bilo kojem trenutku. Sustav za praćenje zdravstvenog stanja oboljelih od dijabetesa sastoji se od programskog rješenja na strani poslužitelja koje je zaduženo za rad s bazom podataka, pohranjivanje, distribuciju, obradu i analizu podataka te posluživanje strane klijenta na zahtjev, web aplikacije kao rješenja kojeg liječnik i pacijent koriste iz web preglednika, mobilne aplikacije orijentirane prema pacijentu te prenosivih uređaja temeljenih na suvremenoj tehnologiji korištenih u svrhu kontinuiranog prikupljanja podataka o mjerenjima pacijenta – senzorske narukvice i glukometra. Najvažniji podaci koje sustav prikuplja od strane pacijenta za praćenje zdravstvenog stanja su podaci o mjerenjima razine glukoze u krvi, doza inzulina, fizičke aktivnosti, dnevnog kretanja i potrošnje kalorija te trajanja i kvalitete pojedinog sna. Liječnicima je omogućeno postavljanje ciljnih vrijednosti kojima pacijenti trebaju težiti te se zdravstveno stanje pacijenta u pojedinom periodu obrađuje i prikazuje kroz odstupanja od ciljnih vrijednosti tog perioda što je popraćeno različitim relevantnim indikatorima koji mogu upozoravati na kritične vrijednosti ili naznačavati da su izmjerene vrijednosti u dozvoljenim rasponima.

Kroz razvijeni sustav pokazala se velika mogućnost povezivanja različitih postojećih tehnologija i njihovo integriranje u jedinstven zdravstveni sustav. Uvođenjem automatiziranog prikupljanja, pravovremene obrade i analize bitnih parametara koji se prikupljaju u ovom sustavu te uz dodavanje i drugih usko vezanih uz dijabetes poput krvnog tlaka, tjelesne mase, tjelesne temperature i ostalih koji su spomenuti u radu, mogao bi se dobiti još bolji pogled na stanje oboljelog od dijabetesa. Radom u timu stručnjaka iz područja računarstva, medicine i drugih vezanih područja mogao bi se razviti sustav prema mjeri koji točno odgovara potrebama liječnika i pacijenata uz praćenje trendova u tehnologiji i maksimalno iskorištavanje tehnoloških resursa koji mogu pomoći oboljelima. Korištenjem naprednih tehnika u prikupljanju i obradi podataka uz uključivanje pacijenata na aktivno sudjelovanje za zajedničko dobro, liječnicima bi bilo jednostavnije razumjeti utjecaj ove bolesti na pojedinog pacijenta te otkriti načine za smanjivanje utjecaja bolesti, a pacijenti bi bili dodatno motivirani na poduzimanje akcija bitnih za vlastito zdravlje.

5. LITERATURA

- [1] What is Diabetes?, National Institute of Diabetes and Digestive and Kidney Diseases, <https://www.niddk.nih.gov/health-information/diabetes/overview/what-is-diabetes>, pristupljeno: 06.06.2017.
- [2] Blood Sugar Level Ranges, Diabetes.co.uk the global diabetes community, http://www.diabetes.co.uk/diabetes_care/blood-sugar-level-ranges.html, pristupljeno: 07.06.2017.
- [3] Global report on diabetes, Svjetska zdravstvena organizacija (WHO, World Health Organization), Francuska, 2016
- [4] T. Poljičanin, L. Smirčić, M. Vinković, V. Kolarić, Šećerna bolest u Republici Hrvatskoj 2005. – 2014., Hrvatski zavod za javno zdravstvo, Zagreb, Republika Hrvatska, 2015
- [5] Short Term Complications, Diabetes.co.uk the global diabetes community, <http://www.diabetes.co.uk/diabetes-complications/short-term-complications.html>, pristupljeno: 07.06.2017.
- [6] Diabetes and Hypoglycaemia, Diabetes.co.uk the global diabetes community, <http://www.diabetes.co.uk/Diabetes-and-Hypoglycaemia.html>, pristupljeno: 08.06.2017.
- [7] Alcohol and Hypoglycemia, Diabetes.co.uk the global diabetes community, <http://www.diabetes.co.uk/alcohol-and-hypoglycemia.html>, pristupljeno: 08.06.2017.
- [8] Diabetic Ketoacidosis, Diabetes.co.uk the global diabetes community, <http://www.diabetes.co.uk/diabetes-complications/diabetic-ketoacidosis.html>, pristupljeno: 08.06.2017.
- [9] Centers for Disease Control and Prevention, Take Charge of Your Diabetes, U.S. Department of Health and Human Services, Atlanta, Sjedinjene Američke Države, 4. izdanje, 2007
- [10] Nerve Damage (Diabetic Neuropathies), National Institute of Diabetes and Digestive and Kidney Diseases <https://www.niddk.nih.gov/health-information/diabetes/overview/preventing-problems/nerve-damage-diabetic-neuropathies>, pristupljeno: 10.06.2017.
- [11] Diabetes and Foot Problems, National Institute of Diabetes and Digestive and Kidney Diseases, <https://www.niddk.nih.gov/health-information/diabetes/overview/preventing-problems/foot-problems>, pristupljeno: 10.06.2017.
- [12] Type 2 diabetes, Mayo Clinic, <http://www.mayoclinic.org/diseases-conditions/type-2-diabetes/symptoms-causes/dxc-20169861>, pristupljeno: 10.06.2017.

- [13] Prediabetes & Insulin Resistance, National Institute of Diabetes and Digestive and Kidney Diseases, <https://www.niddk.nih.gov/health-information/diabetes/overview/what-is-diabetes/prediabetes-insulin-resistance>, pristupljeno: 11.06.2017.
- [14] Type 2 diabetes Diagnosis, Mayo Clinic, <http://www.mayoclinic.org/diseases-conditions/type-2-diabetes/diagnosis-treatment/treatment/txc-20169988>, pristupljeno: 11.06.2017.
- [15] Glycemic index and glycemic load for 100+ foods, Harvard Health Publishing, <http://www.health.harvard.edu/diseases-and-conditions/glycemic-index-and-glycemic-load-for-100-foods>, pristupljeno: 11.06.2017.
- [16] Treatment for Type 2 Diabetes, Diabetes.co.uk the global diabetes community, <http://www.diabetes.co.uk/treatment-for-type2-diabetes.html>, pristupljeno: 12.06.2017.
- [17] Insulin, Medicines & Other Diabetes Treatments, National Institute of Diabetes and Digestive and Kidney Diseases, <https://www.niddk.nih.gov/health-information/diabetes/overview/insulin-medicines-treatments>, pristupljeno: 12.06.2017.
- [18] Calculating Insulin Dose, Diabetes Education Online, Diabetes Teaching Center at the University of California, San Francisco, <https://dtc.ucsf.edu/types-of-diabetes/type2/treatment-of-type-2-diabetes/medications-and-therapies/type-2-insulin-rx/calculating-insulin-dose/>, pristupljeno: 12.06.2017.
- [19] What's a Correction Factor? An Insulin Sensitivity? A ratio?, Mount Sinai Hospital Sinai Health System, <http://www.mountsinai.on.ca/care/lscd/sweet-talk-1/what2019s-a-correction-factor-an-insulin-sensitivity-a-ratio>, pristupljeno: 13.06.2017.
- [20] Banner Good Samaritan Medical Center, Physicians dosing guidelines for subcutaneous insulin, 2007
- [21] Day-to-Day Diabetes Management, NHS Lothian, <http://www.nhslothian.scot.nhs.uk/Services/A-Z/DiabetesService/PaediatricDiabetesHandbook/j.day-to-day-diabetes-management.pdf>, pristupljeno: 13.06.2017.
- [22] Managing Diabetes, National Institute of Diabetes and Digestive and Kidney Diseases, <https://www.niddk.nih.gov/health-information/diabetes/overview/managing-diabetes>, pristupljeno: 13.06.2017.
- [23] Daily Diabetes Management Book, Central Ohio Diabetes Association, http://www.diabetesohio.org/Portals/0/docs/DiabetesLiterature/daily_diabetes_management_diary.pdf, pristupljeno: 14.06.2017.

- [24] Type 2 Diabetes Diet, Everyday Health, <http://www.everydayhealth.com/type-2-diabetes/guide/diet/>, pristupljeno: 14.06.2017.
- [25] Body mass index (BMI) and waist circumference, Health Direct, <https://www.healthdirect.gov.au/body-mass-index-bmi-and-waist-circumference>, pristupljeno: 14.06.2017.
- [26] M. Krošel, L. Švegl, L. Vidmar, D. Dinevski, Empowering Diabetes Patient with Mobile Health Technologies, Mobile Health Technologies - Theories and Applications, (ur.) Wilfred Bonney, InTech publishing, 2016.
- [27] E. Årsand, D. H. Frøisland, S. O. Skrøvseth, T. Chomutare, N. Tatara, G. Hartvigsen, J. T. Tufano, Mobile health applications to assist patients with diabetes: lessons learned and design implications, Journal of diabetes science and technology, sv. 6, br. 5, 2012, str. 1197-206
- [28] MyNetDiary, <http://www.mynetdiary.com/>, pristupljeno: 19.06.2017.
- [29] Glooko, <http://www.glooko.com>, pristupljeno: 19.06.2017.
- [30] Glucosio, <http://www.glucosio.org/>, pristupljeno: 19.06.2017.
- [31] J. Bacon, Introduction to LAMP technology: Explore the open source Web development platform, IBM, 2005
- [32] PHP MVC Framework, WebLiquids, <http://www.webliquids.com/php-mvc-framework-training-chandigarh/>, pristupljeno: 04.02.2018.
- [33] S. McCool, Laravel Starter, Packt publishing, Birmingham, Ujedinjeno Kraljevstvo, 2012
- [34] JavaScript Introduction, MDN web docs, <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>, pristupljeno: 04.02.2018.
- [35] J. Chaffer, K. Swedberg, Learning jQuery, Packt publishing, Birmingham, Ujedinjeno Kraljevstvo, 2007
- [36] Handlebars, <http://handlebarsjs.com/>, pristupljeno: 04.02.2018.
- [37] Accelerometer Basics, Sparkfun, <https://learn.sparkfun.com/tutorials/accelerometer-basics>
- [38] Fitbit Help, <https://help.fitbit.com/>, pristupljeno: 04.02.2018.
- [39] Fitbit Community, <https://community.fitbit.com>, pristupljeno: 05.02.2018.
- [40] An Introduction to OAuth 2, DigitalOcean, <https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2>, pristupljeno: 05.02.2018.
- [41] OAuth 2, Fitbit Dev, <https://dev.fitbit.com/build/reference/web-api/oauth2/#authorization-page>, pristupljeno: 05.02.2018.

- [42] W.M. Lee, Beginning Android Application Development, John Wiley & Sons, Indianapolis, Sjedinjene Američke Države, 2012
- [43] Android, Tutorialspoint, <https://www.tutorialspoint.com/android>, pristupljeno: 05.02.2018.
- [44] Consuming APIs with Retrofit, CodePath, <https://guides.codepath.com/android/Consuming-APIs-with-Retrofit>, pristupljeno: 05.02.2018.
- [45] Introduction to Bluetooth Low Energy, Adafruit, <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt>, pristupljeno: 06.02.2018.
- [46] Bluetooth Low Energy, Android Developer, <https://developer.android.com/guide/topics/connectivity/bluetooth-le.html>, pristupljeno: 06.02.2018.

SAŽETAK

Sustav namijenjen praćenju zdravstvenog stanja oboljelih od dijabetesa temelji se na potrebama, funkcionalnostima i karakteristikama koje su se kroz znanstvena istraživanja pokazale nužnima za implementaciju kvalitetnog i djelotvornog programskog rješenja. Sustav prikazuje mogućnost primjene suvremenih informacijskih i komunikacijskih tehnologija u praćenju i u nadzoru zdravstvenog stanja osoba oboljelih od dijabetesa. Sastoji se od web platforme koja služi kao poslužitelj kojoj je cilj povezati cijeli sustav u cjelinu i omogućiti pristup podacima, web aplikacije koja pruža glavni dio korisničkog sučelja, mobilne aplikacije kod koje se ističe uloga sinkronizacije s glukometrom i integriranih prenosivih uređaja u sustav - Fitbit Charge HR senzorske narukvice te Contour Next One Bluetooth glukometra. Jedan od najvećih ciljeva sustava je, osim mogućnosti naprednog pristupa u praćenju zdravstvenog stanja, u što većoj mjeri olakšati pristup podacima automatskim prikupljanjem primjenom modernih tehnologija. Kroz rad je na temelju praktičnog dijela opisan razvoj i funkcionalnost programskog rješenja, kao i tehnologije korištene u samom razvoju.

Ključne riječi: dijabetes, mobilna aplikacija, praćenje, web aplikacija, zdravstveno stanje.

ABSTRACT

Title: Health indicators monitoring computer system for diabetes patients

System for monitoring the health status of people with diabetes is based on the needs, functionalities and characteristics that have been proven to be necessary for the implementation of a quality and effective software solution through scientific research. The system demonstrates the ability of applying modern information and communication technologies in monitoring and controlling health status of those affected by diabetes. It consists of a web platform that serves as a server whose main goal is to connect entire system as a whole and provide the access to data, web application that provides main part of the user interface, mobile application which highlights the role of synchronization with a glucometer and mobile devices integrated into the system – Fitbit Charge HR sensor bracelet and Countour Next One Bluetooth glucometer. One of the major goals of the system is, apart from the advanced approach in health monitoring, to facilitate the access to patient's data as much as possible by automating the process of data collection. Throughout the thesis, development and functionality of the software solution, as well as the technology used in the development itself, is described based on the practical part of the thesis.

Keywords: diabetes, mobile application, monitoring, web application, health status.

ŽIVOTOPIS

Luka Bartolić rođen je 20. svibnja 1993. u Čakovcu. U Čakovcu stječe osnovnoškolsko obrazovanje u 2. osnovnoj školi Čakovec od 2000. do 2008. godine. Gimnaziju Čakovec, prirodoslovno-matematički smjer, upisuje 2008. godine gdje razrede prolazi s vrlo dobrim i odličnim uspjehom. Nakon završetka srednješškolskog obrazovanja, 2012. godine upisuje Elektrotehnički Fakultet u Osijeku, preddiplomski studij računarstva kojeg završava 2015. godine. Iste godine upisuje diplomski studij računarstva, modul programsko inženjerstvo, na istom fakultetu izmijenjenog naziva u Fakultet elektrotehnike, računarstva i informacijskih tehnologija, kojeg trenutno pohađa.

PRILOZI

Na CD-u:

- a) diplomski rad „Računalni sustav namijenjen praćenju pokazatelja zdravstvenog stanja oboljelih od dijabetesa“ u *.docx* formatu
- b) diplomski rad „Računalni sustav namijenjen praćenju pokazatelja zdravstvenog stanja oboljelih od dijabetesa“ u *.pdf* formatu
- c) izvorni kod praktičnog dijela rada