Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Patricija Brečko

Zasnova EKG podatkovne baze z uporabo tehnologije ElasticSearch

diplomsko delo

univerzitetni študijski program
prve stopnje
računalništvo in informatika

Mentor: doc. dr. Dejan Lavbič

Ljubljana, 2018

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Pri razvoju informacijskih sistemov se pogosto izkaže, da je ozko grlo raven dostopa do podatkov. V praksi se uporabljajo različni pristopi, od enostavnega dostopa in iskanja po datotekah, uporabe relacijskih podatkovnih baz, skupaj z indeksiranjem in tudi uporaba nerelacijskih podatkovnih baz, predvsem zaradi performančnih zahtev in dobre podpore skaliranju. Podobno velja tudi za področje zdravstva, kjer se na različnih ravneh srečujemo z veliko količino podatkov. Če podatki, ki prihajajo iz različnih virov, niso zapisani v standardiziranih oblikah, imamo še dodatne težave. V okviru zdravstva so podatki EKG eno izmed takšnih področij, kjer ni enotnega soglasja o tehnologiji za zapis in dostop do večje količine podatkov. V okviru diplomskega dela predstavite trenutno stanje in podporo hranjenju in iskanju po digitalnih EKG podatkih. Na podlagi identificiranih pomanjkljivosti predlagajte nov pristop obvladovanja EKG podatkov s pomočjo tehnološke rešitve ElasticSearch in uporabo invertiranih indeksov. Predlagan pristop kritično ovrednotite in predstavite njegove prednosti in slabosti s trenutno najbolj pogosto uporabljanim pristopom na tem področju.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
ANTLR	ANother Tool for Language Recognition	Še eno orodje za prepoznavo jezika
ANSI	American National Standards Institute	Ameriški državni inštitut za standarde
CT	Computed Tomography	Računalniška tomografija
DBMS	Database Management System	Sistem za upravljanje podatkovnih baz
DICOM	Digital Imaging and Communications in Medicine	Digitalno slikanje in komunikacija v medicini
EKG	Electrocardiography	Elektrokardiografija
ES	ElasticSearch	Elastično iskanje
FDA	Food and Drug Administration	Zvezna agencija za hrano in zdravila
HL7	Health Level Seven International	Neprofitna organizacija, ki sodeluje pri razvoju mednarodnih standardov v zdravstvu
JSON	JavaScript Object Notation	Objektna notacija JavaScript
MRI	Magnetic Resonance Imaging	Magnetna resonanca
NRT	Near RealTime	V skoraj realnem času
PACS	Picture Archiving and Communication Systems	Sistemi za arhiviranje slik in komunikacijo
PB	Database	Podatkovna baza

QRS	Q wave, R wave, S wave	Q val, R val, S val
SCP-ECG	Standard Communications Protocol for Computer-Assisted ElectroCardio-Graphy	Standardni komunikacijski protokol za računalniško podprto elektrokardiografijo
WFDB	WaveForm DataBase	Podatkovna baza signalov valovne oblike
XML	eXtensible Markup Language	Razširljivi označevalni jezik

Povzetek

Naslov: Zasnova EKG podatkovne baze z uporabo tehnologije ElasticSearch

Avtor: Patricija Brečko

V zadnjem času smo priča izredno hitremu vzponu tehnologije, ki vpliva na vsa področja našega življenja, med drugim tudi na zdravstvo. To področje z zamikom sledi napredku, saj gre pogosto za podatke, občutljive po naravi ali zahtevne za obdelavo, ki vsekakor zahtevajo več previdnosti pri delu z njimi. Vseeno pa bi bilo do veliko izzivov v zdravstvu zanimivo pristopiti s pomočjo najnovejših tehnologij, kar je tudi tema tega diplomskega dela. Problem, ki ga poskusimo rešiti, je zasnova podatkovne baze za hranjenje digitalnih EKG podatkov, ki bi omogočala učinkovito iskanje. Lotimo se ga z uporabo tehnologije ElasticSearch, ki je hkrati podatkovna baza in iskalno orodje. Izdelamo tudi preprost uporabniški vmesnik, ki uporabnikom brez znanja o ElasticSearchu omogoča iskanje po bazi. Svojo rešitev evalviramo na podlagi nekaj uporabniških scenarijev z vidika zdravnika in predstavimo ugotovljene prednosti in slabosti tega pristopa za reševanje danega problema.

Ključne besede: elektrokardiografija, digitalizacija, ElasticSearch, PhysioNet, EKG podatkovna baza.

Abstract

Title: Design of the ECG database using ElasticSearch technology

Author: Patricija Brečko

In the recent years we have witnessed a rapid rise in technology that affects all areas of our lives, including the health care services. Here it is particularly difficult to follow the latest technological trends, as this often involves the use of data which is sensitive in its nature or complex and difficult to process, which makes it especially important to be careful when working with it. Nevertheless, trying new approaches to solve problems in the medical field using the latest technologies might bring positive results, which is the main reason for the topic of this thesis. Our goal was to design a database for storing digital ECG data, which would allow effective search. We aimed to achieve this using the ElasticSearch technology, being both a database management system and a search engine. In order for it to be usable by medical professionals, usually not skilled in database management, we also implemented a simple user interface. In last part of this thesis we also evaluate our solution based on a couple of user scenarios from the doctor's or researcher's point of view and present the advantages and disadvantages of this approach to solve the given problem.

Keywords: electrocardiography, digitalization, ElasticSearch, PhysioNet, ECG database.

Poglavje 1

Uvod

Znano je, da digitalizacija zdravstva prispeva k boljšim rezultatom pri zdravljenju, poenostavi postopke, zmanjša stroške in nasploh olajša delo zdravnikom. Tudi digitalizacija elektrokardiografije prinaša takšne prednosti, poleg tega pa omogoča naprednejše načine obdelovanja podatkov, kot je avtomatska analiza nepravilnosti v delovanju srca, iskanje po velikem številu zapisov in strojno učenje, kar vse prispeva k bolj učinkovitemu zdravljenju. [15]

To delo se ukvarja z implementacijo enega izmed teh načinov obdelovanja podatkov, to je iskanje po velikem številu EKG zapisov. Pogosto bi kardiologom pri delu pomagala baza z EKG zapisi, po kateri bi lahko iskali po določenih lastnostih podatkov (demografski podatki o pacientih, po določenih diagnozah ali načinu preiskave, itd.), s čimer bi si pomagali pri diagnosticiranju in raziskavah.

Edina prosto dostopna EKG baza, ki trenutno to ponuja, je platforma PhysioNet [9]. Ta hrani zelo bogat nabor podatkov, vendar je omejena pri iskanju po njih - omogoča le iskanje po vseh zapisih (ne pa tudi znotraj posamičnega zapisa) in samo po določenih parametrih. Naša naloga je, da razvijemo alternativo PhysioNet-ovi bazi, ki bi omogočala boljše iskanje.

Naloge se bomo lotili z uporabo tehnologije ElasticSearch, ki omogoča hranjenje velikega števila podatkov in učinkovito iskanje. Razvili bomo tudi uporabniški vmesnik, ki bo ponujal dva načina iskanja. Eden bo omogočal

pisanje poizvedb direktno za bazo, drugi pa sestavljanje poizvedb prek preprostega poizvedovalnega jezika, da bodo rešitev lahko uporabljali tehnično bolj ali manj podkovani raziskovalci.

Učinkovitost svoje rešitve bomo preverili prek izvedbe nekaj realnih uporabniških scenarijev, s katerimi bomo simulirali delo zdravnika oz. raziskovalca. Primerjali bomo njegovo uspešnost z uporabo naše rešitve in z uporabo platforme PhysioNet, oboje prek uporabniškega vmesnika.

Nato bomo primerjali še kompleksnejše poizvedovanje po obeh bazah brez uporabniškega vmesnika in tako evalvirali našo rešitev še iz semantičnega vidika (kako kompleksne poizvedbe je mogoče izvesti in kako tehnično usposobljen mora biti uporabnik).

1.1 Pregled področja

Področje digitalne elektrokardiografije je na splošno precej težavno in to iz več razlogov. Na prvem mestu je seveda ta, da se danes v veliki meri še zmeraj uporablja papir kot glavno sredstvo za shranjevanje podatkov oz. se te izvide potem preprosto fotografira in shrani v bazo. Drugi večji problem pa je, da še ne obstaja enoten oz. standardiziran format za shranjevanje digitalnih EKG zapisov, čeprav se v svetu že veliko dela na tem. Imamo nekaj bolj poznanih formatov, ki bodo tudi predstavljeni v tem delu, a v splošnem velja, da ima skoraj vsaka EKG hiša tudi svoj format za EKG zapis, kar potem precej oteži pretvorbo podatkov v neko enotno obliko. Podatke lahko označimo tudi kot nekoliko bolj občutljive narave (preko EKG posnetka lahko med drugim identificiramo osebo), kar je mogoče tudi razlog, da se na spletu težko najde prosto dostopne primere.

Posledično zaradi teh težav precej dela pri razvoju takšne aplikacije povzroči že samo iskanje primernih podatkov in algoritmov za obdelavo različnih formatov. Večino zapisov smo tako vzeli iz platforme PhysioNet. Ta poleg bogate podatkovne baze vsebuje tudi mnogo algoritmov za obdelavo posnetkov, pretvarjanje med formati, pregledovalnik zapisov in uporabniški vmesnik

za iskanje.

1.2 Struktura dela

Delo je strukturirano tako, da najprej v poglavju 2 predstavimo področje elektrokardiografije. To je pomembno, da bolje spoznamo naravo EKG podatkov, kar nam omogoča bolj premišljeno izbiro podatkovne baze za našo aplikacijo in sestavo podatkovnega modela.

V naslednjem poglavju (3) predstavimo bazo PhysioNet in opišemo njihovo rešitev - kako so oblikovali podatke, katero tehnologijo za podatkovno bazo so izbrali in kakšen je njihov vmesnik za iskanje.

V poglavju 4 se nadaljnje osredotočimo na našo rešitev. Razložimo, zakaj smo za našo podatkovno bazo izbrali ElasticSearch, razložimo glavne koncepte pri tej tehnologiji in opišemo, kako smo oblikovali podatke ter iz katerih virov smo jih črpali. Predstavimo še naš uporabniški vmesnik in kako ga uporabljati.

Nato v poglavju 5 evalviramo našo rešitev. Prikažemo pet uporabniških scenarijev iz stališča zdravnika in ocenimo, kako se obnese naša rešitev. Poleg tega iste uporabniške scenarije prikažemo z uporabo iskalnega orodja PhysioNet in primerjamo rezultate. Obe rešitvi evalviramo še iz stališča bolj tehnično usposobljenega uporabnika in preizkusimo en primer iskanja direktno po podatkovnih bazah, kjer primerjamo težavnost pisanja poizvedb in zahtevano predznanje uporabnika.

Nazadnje pa v poglavju 6 povzamemo svoje ugotovitve in razmislimo o možnih razširitvah.

Poglavje 2

Elektrokardiografija

V tem poglavju bomo najprej predstavili osnove elektrokardiografije in kakšne podatke dobimo pri EKG preiskavah. Opisali bomo nekaj pogostejših preiskav, v drugem podpoglavju pa še karakteristike rezultatov, ki so kardiologom najbolj pomembne pri preučevanju EKG zapisov.

Nato bomo v tretjem podpoglavju opisali, na kakšen način se danes digitalizira EKG podatke in kateri so trenutno aktualni problemi na tem področju. Predstavili bomo najpogostejše formate za shranjevanje digitaliziranih EKG podatkov (DICOM, SCP-ECG, HL7 aECG [3]).

Ob pregledu teh formatov in baze PhysioBank bomo predstavili ugotovitev, katere so tiste najbolj pomembne informacije, ki jih mora vsebovati vsak EKG zapis.

2.1 O elektrokardiografiji

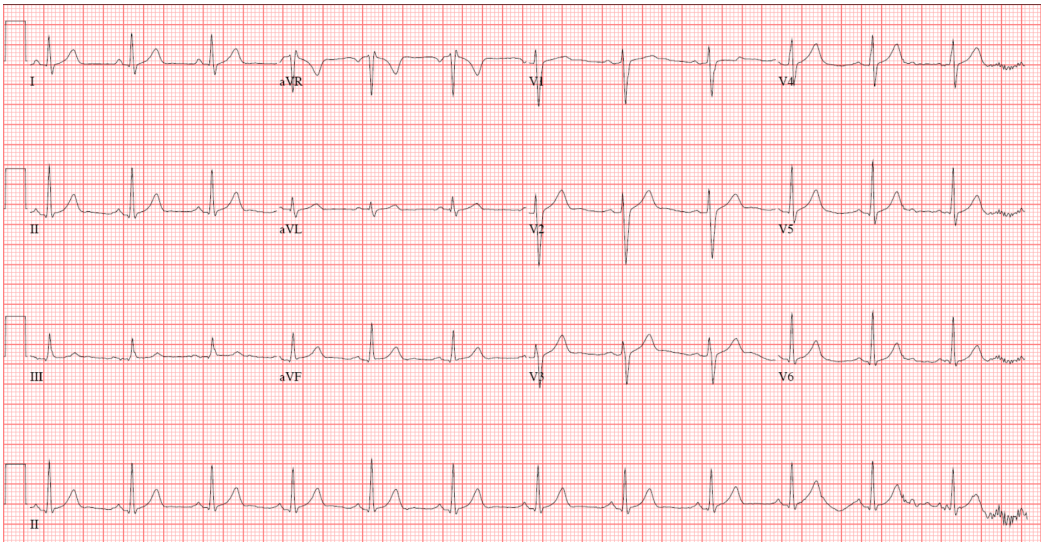
Elektrokardiografija (EKG) je proces snemanja električne aktivnosti srca skozi določeno časovno obdobje [5]. Gre za pogosto in neinvazivno kardiološko preiskavo, njen namen pa je diagnosticirati različne nepravilnosti v delovanju srca, kot so srčni infarkti, hipertrofija, bolezni zaklopk, anomalije v ritmu bitja srca in tako dalje [16].

Pri preiskavi se na kožo namesti elektrode, ki zaznavajo majhne električne

spremembe. Te spremembe se dogajajo zaradi depolarizacije in repolarizacije srčnih mišičnih celic ob vsakem utripu srca [5], to stalno nihanje električne napetosti pa se v EKG zapisu odraža v obliki specifičnih valov, pri katerih opazujemo trajanje, amplitudo in obliko [16].

Med standardno 12-kanalno EKG preiskavo se na telo navadno pritrdi 10 elektrod. Da lahko natančneje opazujemo delovanje različnih delov srca, računamo razlike v električni napetosti med dvema (ali več) točkami na telesu (kjer so posajene elektrode) - tem razlikam pravimo odvodi. Najpomembnejših odvodov je 12: I, II, III, aVL, aVR, aVF, V1, V2, V3, V4, V5 in V6. Ob različnih preiskavah imamo lahko tudi manj odvodov, različice zgoraj naštetih (npr. MLII) ali dodatne odvode (npr. V7, V8, V9) [5].

Rezultate prikazujemo na elektrokardiogramu - grafu električne napetosti (običajno v milivoltih) v odvisnosti od časa (običajno v sekundah) [5]. Elektrokardiogram je najbolj učinkovit način predstavitve rezultatov kardiologu, saj lahko ta že s hitrim pregledom diagnosticira nekatere nepravilnosti. Danes se v zdravstvu še zmeraj v večini uporabljajo elektrokardiogrami, natisnjeni na papirju. Primer papirnatega elektrokardiograma je na sliki 2.1.



Slika 2.1: Elektrokardiogram

Oblika EKG zapisa je odvisna tudi od vrste preiskave, ki je bila izvedena.

Obstaja več kot 10 različnih vrst, med katerimi so ene izmed pogostejših [8]:

Standardni 12 kanalni EKG test Traja okoli 10 sekund in posname srce z 10 elektrodami v 12 odvodih. Običajno se po snemanju natisne na papir, da ga kardiolog lahko takoj pregleda [5].

Holter EKG test Namenjen odkrivanju motenj, ki jih z običajnim EKG posnetkom ne uspemo zaslediti, saj gre za najmanj 24-urno snemanje. Poleg tega mora pacient voditi tudi dnevnik svojih aktivnosti. Odvoda sta ponavadi dva ali trije. Posnetek sproti analizira računalnik in ga vrne v elektronski obliki po koncu snemanja [10].

Obremenitveni test Pacient med snemanjem izvaja neko športno aktivnost, recimo vožnjo kolesa ali hojo po tekalni stezi. Test traja med 7 in 12 minut, snema pa se vseh 12 kanalov [6].

2.2 Oblika podatkov

Pri izvornih in neobdelanih EKG podatkih gre torej za dolg seznam vrednosti električnih napetosti ob določenih časih. Pri nekaterih zapisih je časovni interval med vzorci vrednosti konstantno isti, pri nekaterih pa so različni. Frekvence snemanja so navadno nekje med 100 in 1000 vzorcev na sekundo. [5]

Strokovnjaki ne preučujejo posameznih vrednosti, temveč oblike krivulje na grafu. Opazujejo oblike krivulje pri posameznih udarcih srca na enem ali na več odvodih hkrati, prav tako so pomembne tudi dolžine časovnih presledkov med posameznimi utripi, čemur pravimo RR intervali. Ob preiskavi se navadno izračuna povprečna vrednost RR intervala (iz česar izračunamo hitrost bitja srca - število utripov na minuto), najkrajši in pa najdaljši RR interval. [16]

Pri kratkih posnetkih ročno pregledovanje zgornjih lastnosti ne predstavlja posebnih težav, če pa so posnetki daljši (sploh recimo pri Holter EKG testu, ki lahko traja več dni), pa je takšen način preveč zamuden. Ob takšnih vrstah preiskav elektrokardiograf ponavadi že sam sproti označuje (anotira)

zanimivejše dele na posnetku, da se potem zdravnik posveti samo tem delom. [10]

Pri nekaterih preiskavah so pomembne tudi dejavnosti, ki jih sam elektrokardiograf ne more posneti, npr. kaj je pacient počel v času neobičajnih EKG posnetkov (če gre za npr. Holterjev test). V takšnih primerih pacient običajno vodi dnevnik aktivnosti in ga potem ob koncu preiskave predloži zdravniku skupaj s posnetki iz elektrokardiografa [10]. Včasih se poleg EKG-ja spremljata še dihanje in krvni pritisk, saj oba vplivata na rezultate [3].

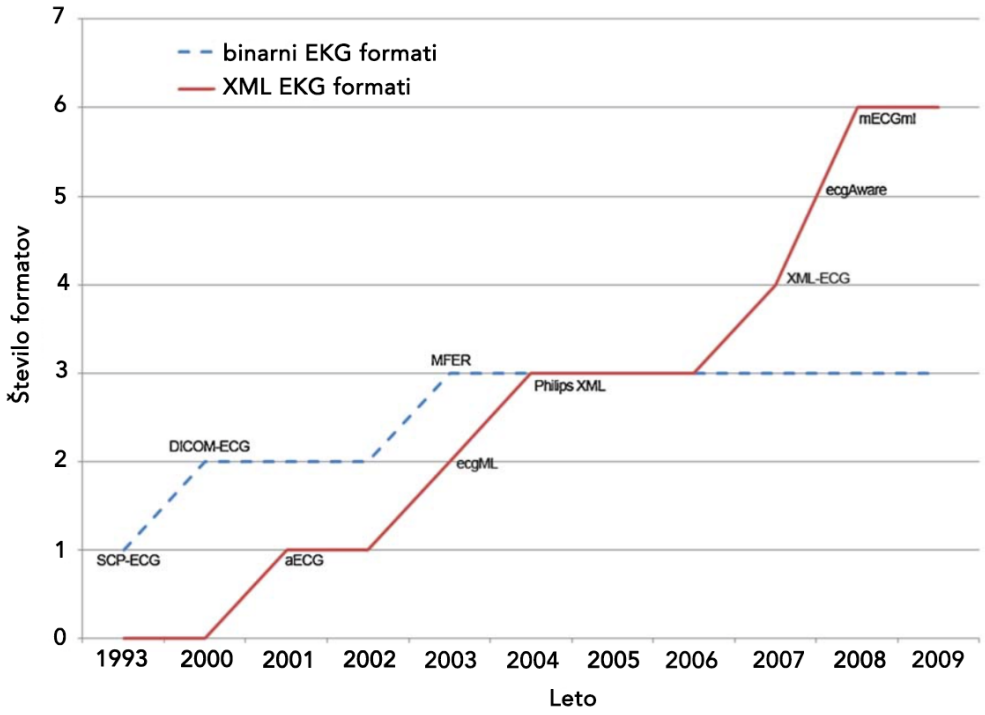
2.3 Digitalizacija EKG

Znano je, da digitalizacija zdravstvenih zapisov pomaga k boljšim rezultatom pri zdravljenju, poenostavi postopke, zmanjša stroške in nasploh olajša delo zdravnikom - tako je tudi pri elektrokardiografiji [15]. V mnogih zdravstvenih organizacijah (tudi v Sloveniji) se danes še zmeraj EKG zapise tiska na papir, vendar gre razvoj počasi v smer elektronskih zapisov. Od samega začetka digitalizacije elektrokardiografije pa vse do danes se mnenja o tem, kateri je najustreznejši format za shranjevanje digitalnih EKG podatkov, močno delijo.

Nekateri zagovarjajo bolj interoperabilne formate, torej da poleg shranjevanja EKG podatkov podpirajo še shranjevanje podobnih medicinskih posnetkov (npr. MRI). To po eni strani olajša branje različnih vrst zapisov, vendar hkrati pelje tudi v veliko kompleksnost formata, zato drugi zagovarjajo, da naj obstaja specifičen format samo za EKG. Pri tem so šli nekateri večji proizvajalci elektrokardiografov še korak dlje in določili več različnih formatov za EKG, ki so prilagojeni za različne naprave, a zato manjši in preprostejši. [3]

Dilema obstaja tudi pri vprašanju, ali naj bodo podatki shranjeni v binarni ali tekstovni obliki. Na začetku je bila pogostejša binarna oblika, saj se je s tem prihranilo na prostoru, a danes prostor ni več takšna omejitev. Čedalje bolj priljubljeni zato postajajo tekstovni formati, saj imajo poleg

berljivosti tudi druge prednosti, npr. omogočajo strojno učenje na podatkih. Na sliki 2.2 vidimo, da čeprav je bilo na začetku več binarnih formatov, se je njihova rast ustavila, še vedno pa je naraščalo število tekstovnih formatov. [3]



Slika 2.2: Primerjava rasti števila binarnih EKG formatov (črtkana črta) in tekstovnih EKG formatov (polna črta)

Posledično danes ni nekega standardiziranega formata za shranjevanje, kar je sicer velika težava na tem področju. V tem delu bomo zato predstavili tri izmed bolj znanih formatov - DICOM, SCP-ECG in HL7 aECG. [3, 17]

2.3.1 DICOM

DICOM format je danes standard za shranjevanje in prenašanje medicinskih slik. V zadnjih 50 letih se je razvilo več radioloških preiskav (CT, MRI), zato je nastala potreba po skupnem digitalnem formatu za radiografske slike in pripadajoče metapodatke za različne naprave - tiskalnike, skenerje, strežnike

ter PACS sisteme. V ta namen so leta 1993 razvili format DICOM, ki je postal evropski standard leta 1995. [3]

Na začetku je format podpiral samo radiografske slike, a se je od takrat precej razširil. Leta 2000 je bil predstavljen tudi format DICOM-WS 30 (ali DICOM-ECG), ki podpira shranjevanje medicinskih podatkov v obliki grafov, zato se ga uporablja za shranjevanje EKG-ja, pa tudi podatkov o krvnem pritisku, zvoku, itd. [3]

Attribute name	Tag	Description	12-lead ECG constraints
Waveform sequence	(5400,0100)	The number of waveform sequence items.	Between 1 and 5. For example, the number 5 would be assigned to imitate the standard 4 × 3 + 1 12-lead ECG lead layout. Sequence 1 or multiplex group 1 would represent leads: I, II, II Multiplex group 2: aVR, aVL, aVF. Multiplex group 3: V1, V2, V3. Multiplex group 4: V4, V5, V6. Multiplex group 5: Would include lead II as the rhythm strip.
Waveform originality	(003A,0004)	Enumeration: ORIGINAL DERIVED	None specified.
Number of channels	(003A,0005)	Number of channels for a multiplex group.	Number of channels should not exceed 13. This includes the 12 leads and the one rhythm strip.
Number of samples	(003A,0010)	Number of samples per channel.	Should be less or equal to 16,384.
Sampling frequency	(003A,001A)	Frequency in Hz.	Should be between 200 Hz and 1000 Hz.
Channel label	(003A,0203)	Text label that represents the channel.	None specified.

Slika 2.3: Struktura formata DICOM-ECG

Prednosti	**Slabosti**
možnost shranjevanja podatkov iz različnih modalitet - lažji razvoj enotne platforme	zapletenost formata
velika skupnost uporabnikov in podpornikov	binaren, zato ni berljiv za človeka
del PACS sistema	kompresija podatkov ni podprta

Tabela 2.1: Prednosti in slabosti formata DICOM-ECG

Podatki so shranjeni v binarni obliki, DICOM pa zaradi tako različnih možnosti uporabe velja za precej zapleten format (struktura na sliki 2.3). Nekatera podjetja v industriji so ga posvojila, vendar v manjšini. Še vedno je težko najti prosto dostopen DICOM-ECG pregledovalnik. [3]

Njegove prednosti in slabosti so povzete v tabeli 2.1.

2.3.2 SCP-ECG

SCP-ECG format je bil razvit v letu 1993, od leta 2005 dalje pa je tudi uraden evropski standard za shranjevanje in prenos EKG zapisov. Namenjen je izključno EKG-ju, leta 2002 ga je posvojil tudi evropski konzorcij OpenECG, ki se je zavzemal za interoperabilnost v digitalni elektrokardiografiji. [3, 17]

Name	Description	Usage
Checksum	Cyclic Redundancy Check (CRC) developed with reference to the CCITT standard. *This section contains 2 bytes.*	Mandatory
File Size	This section contains the size of the entire ECG file in bytes. *This section contains 4 unsigned bytes.*	Mandatory
Section 0	Pointer section (table of contents). *This can have a variable number of bytes.*	Mandatory
Section 1	Header section, i.e. patient demographics. *This can have a variable number of bytes.*	Mandatory
Section 2	Encoding of ECG data, i.e. Huffman tables. *This can have a variable number of bytes.*	Optional
Section 3	Lead definition. *This can have a variable number of bytes.*	Optional
Section 4	QRS locations. *This can have a variable number of bytes.*	Optional
Section 5	Reference beat data. *This can have a variable number of bytes.*	Optional
Section 6	Rhythm or residual data. *This can have a variable number of bytes.*	Optional
Section 7	Global measurements. *This can have a variable number of bytes.*	Optional
Section 8	Diagnosis. *This can have a variable number of bytes.*	Optional
Section 9	Manufacturer specific diagnostic information. *This can have a variable number of bytes.*	Optional
Section 10	Lead measurements. *This can have a variable number of bytes.*	Optional
Section 11	Universal statement codes. *This can have a variable number of bytes.*	Optional

Slika 2.4: Struktura formata SCP-ECG

Prednosti	**Slabosti**
majhna velikost datotek (zaradi binarne oblike in kompresije)	zapletenost formata zaradi kompresijskih tehnik
možnost shranjevanja določenih anotacij	možnosti napak pri dekodiranju
	binaren, zato ni berljiv za človeka
	podpira samo shranjevanje podatkov iz standardnega 12-kanalnega testa (in do okoli 60 sekund)

Tabela 2.2: Prednosti in slabosti formata SCP-ECG

V SCP-ECG formatu so podatki v binarni obliki, poleg surovih podatkov

o signalu pa omogoča tudi shranjevanje nekaterih anotacij, kot so oznake QRS intervalov in tekstovno diagnozo. Podatki so kompresirani, zaradi česar so SCP datoteke zelo majhne (do 10x manjše od DICOM formata in do 40x manjše od tekstovnih formatov), vendar pa imajo zato tudi zapleteno strukturo (na sliki 2.4). [3, 17]

Prednosti in slabosti SCP-ECG formata so povzete v tabeli 2.2.

2.3.3 HL7 aECG

HL7 aECG (*annotated ECG*) je prvi EKG format v XML obliki. Razvila ga je organizacija HL7 po naročilu Zvezne agencije za hrano in zdravila (FDA) leta 2001. HL7 se zavzema za interoperabilnost medicinskih zapisov in razvija standarde za shranjevanje in upravljanje zdravstvenih podatkov. [3]

Prednosti	**Slabosti**
XML oblika, kar omogoča berljivost s preprostim urejevalnikom besedil	velikost datotek
zaradi tekstovne oblike primeren za podatkovno rudarjenje in iskanje po določenih lastnostih	specifičen za FDA (XML shema vsebuje veliko elementov povezanih s kliničnim testiranjem zdravil)
lažji za obdelavo za razvijalce kot binarne datoteke	

Tabela 2.3: Prednosti in slabosti formata HL7 aECG

V procesu potrjevanja novih zdravil so v končni fazi zdravila testirana tudi na ljudeh. Med temi testiranji se pogosto snema EKG in rezultate se potem posreduje FDA. V preteklosti je FDA prejemala veliko število teh rezultatov v različnih formatih (pogosto skenirani zapisi v papirnati obliki) in upravljanje s temi datotekami je postalo preveč nepregledno. Prav tako ni

bilo konsistence med danimi podatki, zato je FDA naročila izdelavo novega formata. Ta format je potem postal standard pri Ameriškemu državnemu inštitutu za standarde (ANSI) leta 2004. FDA zdaj sprejema EKG zapise samo v aECG formatu. [3]

Ker je v XML obliki, je HL7 aECG že po naravi precej nekompakten format, primeren bolj za manjše velikosti zapisov, kar je za rabo FDA vseeno dovolj. [8]

Prednosti in slabosti formata so povzete v tabeli 2.3.

2.4 Oris strukture podatkov

Pri EKG-ju gre torej za dokaj zapletene in včasih obsežne podatke, ki se precej razlikujejo od zapisa do zapisa.

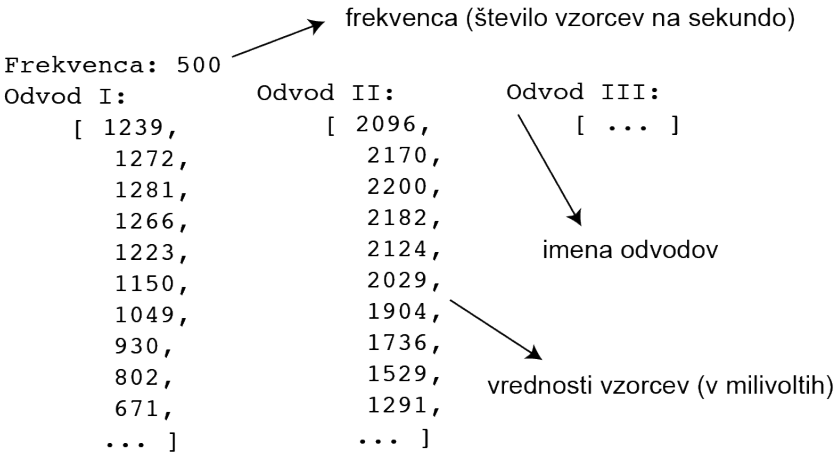
Vsem zapisom je skupno, da vsebujejo sezname vrednosti posnetih napetosti (v minivoltih) ob določenih časih. Ti časi so v nekaterih primerih podani za vsak posnet signal posebej (pogosto pri digitalizaciji papirnatih elektrokardiogramov - slika 2.6), včasih pa ima posnetek določeno frekvenco snemanja in enake časovne razmike med signali, zato v tem primeru ni treba podajati časa za vsak signal posebej, temveč podamo le frekvenco snemanja (slika 2.5). Nujno pa je pri vsakem seznamu vrednosti treba povedati, za kateri odvod gre.

Zapisi navadno vsebujejo tudi metapodatke, kot so podatki o pacientu (identifikacijska številka, starost, spol, itd.) in podatki o samem snemanju (datum zajema posnetka, naprava uporabljena za snemanje itd.)

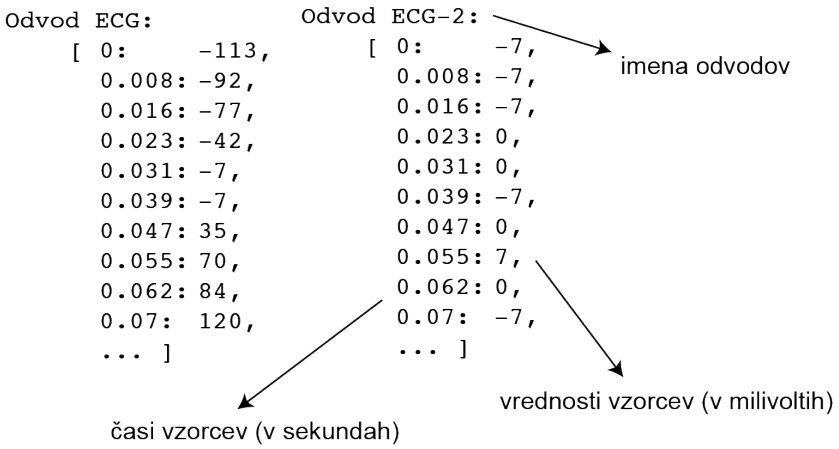
Zelo uporabno za zdravnike je, če zapisi vsebujejo tudi anotacije. Uporablja se jih veliko, če jih naštejemo le nekaj:

- pozicije ST segmentov,
- pozicije in trajanje QRS kompleksov,
- dolžine RR intervalov,

- oznake posameznih utripov,
- opisi ritma bitja,
- diagnoza.



Slika 2.5: Oris strukture podatkov, ki imajo podano frekvenco snemanja



Slika 2.6: Oris strukture podatkov, ki imajo podane čase snemanja

Poglavje 3

PhysioNet

To poglavje je namenjeno opisu platforme PhysioNet, ki omogoča prost dostop do obsežne zbirke posnetkov fizioloških signalov (PhysioBank) in povezane odprtokodne programske opreme (PhysioToolkit). [9]

V prvem podpoglavju bomo torej predstavili platformo PhysioNet. Drugo podpoglavje je namenjeno opisu strukture EKG podatkov v PhysioBank bazi. V tretjem podpoglavju pa si bomo ogledali še različne načine, kako lahko iščemo po PhysioBank bazi.

3.1 Platforma PhysioNet

Projekt PhysioNet so leta 1999 začeli raziskovalci na MIT z namenom spodbujanja raziskav na področju biomedicinskih in fizioloških signalov. Ponujajo prost dostop do podatkov in programske opreme, prav tako tudi spodbujajo druge raziskovalce k prispevanju. Platformo uporablja mnogo biomedicinskih in kliničnih raziskovalcev iz akademskih krogov in industrije, prav tako pa tudi fiziologi, matematiki, računalničarji, študenti in drugi. [9]

Ena izmed treh glavnih komponent platforme PhysioNet je PhysioBank, to je zbirka digitalnih posnetkov različnih preiskav, med katerimi je tudi EKG. PhysioBank vsebuje več kot 90.000 posnetkov (oz. več kot 4 terabajte digitaliziranih podatkov), razporejenih po okoli 80 podatkovnih bazah. Po-

datkovna baza je v tem kontekstu samo zbirka posnetkov v obliki preprostih binarnih datotek. Vsebujejo zapise tako zdravih pacientov kot pacientov z različnimi srčnimi obolenji, dolžine zapisov pa so od nekaj sekund do tudi več kot en dan. Anotirani so bili s strani zdravnika ali preverjenega algoritma. [9]

Pri PhysioNet-u so se odločili, da za hranjenje podatkov ne bodo uporabljali nobenega izmed standardnih formatov, temveč so razvili svojega - ta se imenuje WFDB (WaveForm Database) format.

3.2 Struktura EKG podatkov

En EKG zapis v WFDB formatu identificiramo prek imena datoteke in končnice. Vse datoteke z istim imenom pripadajo istemu zapisu, končnica pa pove, katero izmed komponent zapisa predstavlja [8, 13]:

***header file* - zglavna datoteka (.hea)** kratka tekstovna datoteka, ki opisuje metapodatke posnetka (ime posnetka, format shranjevanja, število in tip odvodov, frekvenca vzorčenja, trajanje posnetka, začetni čas in ostale klinične podatke, ki so na voljo)

***data file* - datoteka s podatki (.dat)** binarna datoteka, ki vsebuje celoten zapis signalov

***annotation file* - anotacijska datoteka (.atr)** binarna datoteka z anotacijami EKG zapisa (opisujejo neko lastnost posnetka in čas, kdaj se je zgodila). Lahko jih je več, vsaka hrani svojo vrsto anotacij.

S temi podatki lahko delamo s pomočjo programske opreme PhysioToolkit. Ta programska oprema vključuje mnogo različnih knjižnic, ki nam (med drugim) omogočajo:

- prenašanje podatkov prek HTTP protokola,
- prenašanje celih posnetkov ali pa le delov le-teh, če so preveliki (recimo če je nek posnetek dolg 40 ur, nas pa zanima samo določenih 5 minut),

- pretvarjanje med različnimi formati,
- procesiranje signalov v različnih jezikih (C, Java, Matlab, Python), itd.

Zaradi velike količine podatkov v tem formatu, široke skupnosti in ogromno razvite programske opreme za obdelavo, je WFDB domnevno trenutno eden izmed najboljših formatov za shranjevanje EKG. [8]

3.3 Iskanje

V kontekstu, kjer je podatkovna baza samo zbirka mnogih datotek, je iskanje počasno, saj je treba odpirati in prebirati datoteko za datoteko.

To težavo so pri PhysioNet-u rešili tako, da so naredili indeksno datoteko, ki hrani imena vseh zapisov in njihove lastnosti, po katerih lahko iščemo. PhysioBank indeks je trenutno 138 megabajtov velika tekstovna datoteka. Vsaka vrstica opisuje eno lastnost, odvod ali anotacijsko datoteko enega izmed zapisov - v indeksni datoteki je okoli 860,000 vrstic, razporejene pa so po abecednem vrstnem redu. Izgled indeksa je prikazan na sliki 3.1. Ne vsebuje informacij o dogodkih znotraj posameznega zapisa in njihovih lokacijah, tako da po teh parametrih ne moremo iskati. [11]

```
edb/e0103       Info1   Mixed angina
edb/e0103       Info2   1-vessel disease (RCA)
edb/e0103       Meds1   nitrates, diltiazem
edb/e0103       Info3   Recorder type: ICR 7200
edb/e0103       AgeSex  62      M
edb/e0103       ECG1    V4      250     200 adu/mV      7200
edb/e0103       ECG2    MLIII   250     200 adu/mV      7200
edb/e0103       AnnR1   atr     250     7336    7200    0-7200
edb/e0103       AnnR1   atr/(N  250     9       5243
edb/e0103       AnnR1   atr/N   250     7212    7199    0-7200
edb/e0103       AnnR1   atr/s   250     15      6098    859-6956
edb/e0103       AnnR1   atr/(VT 250     2       6
edb/e0103       AnnR1   atr/V   250     82      5069    1892-6961
edb/e0103       AnnR1   atr/(B  250     6       58
edb/e0103       AnnR1   atr/F   250     2       574     2132-2706
edb/e0103       AnnR1   atr/~   250     8       3613    2682-6295
```

Slika 3.1: PhysioBank indeks

3.3.1 Vmesnik PhysioBank Record Search

Vmesnik za iskanje, ki ga PhysioNet ponuja, se imenuje PhysioBank Record Search [12]. Omogoča sestavljanje poizvedb za iskanje po vseh bazah fizioloških zapisov, iskanje v ozadju pa temelji na indeksni datoteki.

Vmesnik je prikazan na sliki 3.2. Izbiramo lahko parameter, po katerem želimo iskati (*Subject*), kakšne vrednosti naj bo (*Value*) in kakšna mora biti relacija med njima (*Relationship*). Več takih zahtev pa lahko skupaj združujemo z operacijama "in" (*And*), "ali" (*Or*) in "ne" (*Not*). Rezultat poizvedbe je vedno seznam datotek, ki ustrezajo danim kriterijem.

Slika 3.2: PhysioBank Record Search

Paramentri pri EKG zapisih, po katerih lahko iščemo, so:

- ime datoteke (id),
- starost,
- spol,
- diagnoza,
- info (ostali metapodatki),
- zdravilo,

- anotator (človek ali računalnik),
- tip anotacij,
- dolžina zapisa,
- odvodi,
- frekvenca.

(Drugi parametri, ki so še na voljo, so za iskanje drugih fizioloških posnetkov.)

3.3.2 Paket pbsearch (*pbsearch software package*)

Paket pbsearch [11] vsebuje izvorno kodo za spletnega odjemalca in strežnik, ki sta v ozadju orodja PhysioBank Record Search, prav tako pa še odjemalca v obliki ukazne vrstice, iskalni pogon (search engine) in nekaj vtičnikov. Za delovanje potrebujemo kopijo indeksne datoteke. Paket je namenjen tistim uporabnikom, ki želijo iskalnik prilagoditi svojim potrebam.

Poglavje 4

Implementacija z uporabo ElasticSearch podatkovne baze

V prvem delu tega poglavja bomo pregledali, kakšne lastnosti mora imeti naša podatkovna baza, da bo primerna za hranjenje podatkov, opisanih v poglavju 2. Nato bomo še razložili, zakaj se nam je ElasticSearch zdel primerna izbira.

Naslednje podpoglavje bomo namenili opisu osnovnih konceptov in pojmov tehnologije ElasticSearch ter kakšne so iskalne poizvedbe.

V tretjem podpoglavju bomo predstavili naš predlagan podatkovni model, v četrtem pa katere podatke smo uporabili za testiranje in s katerimi algoritmi smo jih obdelali.

V petem podpoglavju bomo predstavili še uporabniški vmesnik, ki smo ga izdelali za poenostavljeno komunikacijo z bazo.

4.1 Izbira podatkovne baze

Glavne zahteve, ki jim mora čim bolje zadostovati naša baza, so torej:

1. veliko iskanja po vseh poljih in po veliki količini podatkov,
2. izvajanje tudi zahtevnejših poizvedb,
3. občasno vstavljanje novih podatkov,

4. redko spreminjanje (ali brisanje) obstoječih podatkov, se jih bo pa občasno dopolnjevalo - npr. dodajanje anotacij na nek zapis,

5. podatki si bodo med seboj zelo raznoliki (na primer, nek EKG zapis ima lahko samo odvod I in starost pacienta, nek drug pa samo odvod II in spol pacienta),

6. z dodajanjem novih anotacij dodajamo nove oblike podatkov - shema podatkovnega modela se spreminja.

Odločili smo se, da ne bomo uporabili relacijske baze, saj že v literaturi naletimo na nekaj argumentov, zakaj takšna baza ne bi bila najbolj primerna. Pri razvoju platforme PhysioBank so se rajši odločili za zbirko binarnih datotek namesto relacijske baze iz naslednjih razlogov [13]:

- Zapisov ni tako veliko in posamezni zapisi so zelo dolgi - večina nekaj megabajtov, nekateri pa tudi do enega gigabajta.

- Z orodji za obdelavo WFDB zapisov lažje izvajajo kompleksnejše poizvedbe, kot bi jih z relacijsko bazo in SQL.

Na tem mestu je treba sicer omeniti, da je na izbiro binarnega formata za shranjevanje najbrž vplivala tudi želja po prihranku prostora, kar pa danes ni več tako bistvenega pomena, saj so diski že precej bolj cenovno dostopni. Poleg tega hranjenje podatkov v tekstovni obliki prinaša še druge prednosti, kot je možnost iskanja po podatkih, berljivost in pa možnost podatkovnega rudarjenja. Iz teh razlogov smo se odločili za tekstovni format.

Izmed nerelacijskih PB se nam je najbolj primeren zdel ElasticSearch (ES). Ena izmed glavnih prednosti te baze je ravno implementacija učinkovitega iskalnega pogona, poleg tega pa še skalabilnost, kar zadošča potrebam prvih dveh zahtev. Njegova prednost je tudi, da hrani podatke v obliki dokumentov, ki imajo fleksibilno strukturo, zato nam ne bo ostajalo kup praznih polj, poleg tega pa lahko poljubno dodajamo nova polja in indekse, kar ustreza naši potrebi po dodajanju novih anotacij.

Pomisleki, na katere smo naleteli ob izbiri ES, so bili predvsem, da je kompleksnost poizvedovanja do neke mere vseeno omejena (treba je najti kompromis med podvojevanjem podatkov in omejenostjo poizvedb) in pa nekoliko balasten poizvedovalni jezik.

4.2 ElasticSearch

ElasticSearch je odprtokodna porazdeljena nerelacijska podatkovna baza in iskalno orodje. Osnovan je tako, da omogoča visoko skalabilnost in dobre performance, prav tako pa zagotavlja NRT (Near Realtime) iskanje - to pomeni, da je latenca med indeksiranjem dokumenta in tem, da lahko poizvemo po njem v bazi, približno 1 sekunda. Iskalno orodje temelji na knjižnici Apache Lucene [2] in tako omogoča tudi iskanje po celotnem besedilu. Vsako polje je indeksirano, uporablja pa se princip invertiranega indeksa. ES je samostojen strežnik, s katerim komuniciramo prek RESTful API-ja. Privzeta številka vrat, ki jih v ta namen uporablja, je 9200. [7]

4.2.1 Osnovni pojmi

dokument (*document*) Je najmanjša podatkovna enota, po kateri lahko poizvedujemo. Sestavljen je iz polj, ki ima vsako svoj tip (tekst, število, datum, lahko pa tudi tabela ali poddokument), podan pa je v JSON formatu. Vsak dokument ima lahko različna polja, prav tako struktura ni fiksna in jo lahko naknadno spreminjamo. Vsak dokument pripada enemu indeksu. Je nekako ekvivalent vrstici v relacijski podatkovni bazi. **Indeksiranje dokumenta** pomeni vstavljanje dokumenta v bazo. [7]

indeks (*index*) Zbirka dokumentov s podobnimi karakteristikami, po kateri lahko iščemo - npr. imamo lahko indeks za podatke o strankah, indeks za katalog izdelkov in indeks za naročila. Definiran je z imenom, na katerega se sklicujemo ob indeksiranju, iskanju, posodabljanju

in brisanju dokumentov. Je nekako ekvivalent tabeli v relacijski bazi. Lahko imamo neomejeno število indeksov. [7]

gruča (*cluster*) Gruča je zbirka enega ali več vozlišč (strežnikov), ki skupaj tvorijo celotno podatkovno bazo in hranijo vse podatke ter omogočajo indeksiranje in iskanje po vseh vozliščih. Vsako vozlišče je ekvivalentno in sposobno usmerjati poizvedbe po ostalih vozliščih, sprejeti delne rezultate in jih združiti v odgovor. [7]

črepinja (*shard*) Indeks lahko potencialno hrani ogromno količino podatkov, ki presežejo fizične omejitve strežnika. Zato nam ES omogoča, da indeks razdelimo na več kosov - črepinj. Te črepinje so potem porazdeljene po različnih vozliščih, kar omogoča skaliranje baze in vzporedno izvajanje operacij, kar tudi izboljša performance. Vse skupaj je za uporabnika transparentno, saj poizvedbe izvaja na enak način, kot če bi bil indeks na enem vozlišču. [7]

kopija (*replica*) V namen odpornosti na odpoved ali nepravilno delovanje posameznih vozlišč ES podpira izdelovanje kopij posameznih črepinj (zato je pomembno, da se dve kopiji iste črepinje ne nahajata na istem vozlišču). Prav tako izboljša performance, saj lahko poizvedbe izvajamo paralelno na več kopijah. [7]

invertiran indeks (*inverted index*) Kot relacijske baze pogosto dodajo indeks (npr. indeksno B-drevo) določenim stolpcem, da izboljšajo hitrost vračanja podatkov, ES v ta namen uporablja stukturo imenovano invertiran indeks. Invertiran indeks v bistvu pomeni, da za vsak niz obstaja seznam ID-jev dokumentov, v katerih se ta nahaja. [7]

4.2.2 Primer poizvedbe

Poizvedbo na ES strežnik lahko naredimo na dva različna načina. Lahko jo izvedemo z uporabo izključno URI, v katerem podamo tudi vse parametre.

Na ta način ne moremo izkoristiti vseh možnosti iskanja, ki jih ES ponuja, je pa poizvedba veliko bolj preprosta. Primer je na odseku kode 4.1. [7]

```
GET /twitter/_search?q=user:kimchy
```

Odsek kode 4.1: Iskanje po vseh dokumentih znotraj indeksa “twitter“

Drugi način za poizvedovanje pa je uporaba domensko specifičnega jezika (DSL), ki ga ponuja ES. Poizvedbe so v obliki hiearhičnega drevesa, pisati pa jih je treba v JSON formatu. Na odseku kode 4.2 je ista poizvedba kot na odseku kode 4.1, le v DSL obliki. [7]

```
GET /twitter/tweet/_search
{
    "query" : {
        "term" : { "user" : "kimchy" }
    }
}
```

Odsek kode 4.2: Iskanje po vseh dokumentih znotraj indeksa “twitter“

V obeh primerih poizvedovanja dobimo odgovor v JSON obliki (odsek kode 4.3).

```
{
    "took": 1,
    "timed_out": false,
    "_shards":{
        "total" : 1,
        "successful" : 1,
        "skipped" : 0,
        "failed" : 0
    },
    "hits":{
        "total" : 1,
        "max_score": 1.3862944,
        "hits" : [
            {
```

```
                "_index" : "twitter",
                "_type" : "tweet",
                "_id" : "0",
                "_score": 1.3862944,
                "_source" : {
                    "user" : "kimchy",
                    "message": "trying out ElasticSearch",
                    "date" : "2009-11-15T14:12:12",
                    "likes" : 0
                }
            }
        ]
    }
}
```

Odsek kode 4.3: Primer odgovora na poizvedbo

4.3 Predlog našega podatkovnega modela

Za prototip naše aplikacije smo si zamislili podatkovni model s štirimi indeksi: indeks z meta podatki, indeks s podatki o signalih, indeks z anotacijami hitrosti srčnega utripa in pa indeks z anotacijami oznak srčnih utripov. Če bi uporabnik želel dodati novo vrsto anotacij, bi dodal še en indeks. V praksi bi pričakovali, da bi bilo anotacijskih indeksov precej več.

V splošnem pri ES velja pravilo, da naj posamezen dokument vsebuje vse informacije, po katerih se bo poizvedovalo. To s takšnimi podatki ni mogoče, saj so tako obširni, da se bo posamezen zapis razpredal čez mnogo dokumentov. Prednost tega pristopa se izkaže pri poizvedovanju znotraj enega dokumenta, saj res dobimo le tiste dele zapisa, ki nas zanimajo. Slabost pa je v tem, da bomo morali občasno izvesti več poizvedb v bazo, saj ES ne omogoča stikov med indeksi. To bi lahko bila pri poizvedbah nad veliko anotacijami precejšnja težava.

4.3.1 Indeks z meta podatki

Indeks z meta podatki smo poimenovali `record_info_data`. Polja, ki se pojavljajo v teh dokumentih, so: ID posnetka (`record_id`), ID pacienta (`patient_id`), starost pacienta (`patient_age`), spol pacienta (`pacient_sex`), število vseh vzorcev signala (`number_of_all_samples`), časovni interval med vzorci (`sample_time_interval`), frekvenca (`frequency`), datum snemanja (`acquisition_date`), dolžina posnetka (`length`), odvodi (`leads`) in dodane anotacije (`annotations`).

Primer dokumenta v tem indeksu je na odseku kode 4.4.

```
{
    "record_id": "74404",
    "patient_id": "patient_123",
    "patient_sex": "M",
    "patient_age": 40,
    "number_of_all_samples": 5000,
    "sample_time_interval": 0.002,
    "frequency": 500,
    "acquisition_date": 2015-02-03,
    "length": 10,
    "leads":
        [
            "I",
            "II",
            "III",
            "aVR",
            "aVL",
            "aVF",
            "V1",
            "V2",
            "V3",
            "V4",
            "V5",
            "V6"
        ],
```

```
    annotations: ["heart_rate"],
    diagnosis: "arrhythmia"
}
```

Odsek kode 4.4: Primer dokumenta iz indeksa `record_info_data`

4.3.2 Indeks z neobdelanimi podatki

Indeks z neobdelanimi podatki smo poimenovali `record_raw_data`. Polja, ki se lahko pojavijo v teh dokumentih, so: ID posnetka (`record_id`), zaporedno število dokumenta (`record_number`), odvod (`lead`), število vzorcev v tem dokumentu (`number_of_samples`), začetni čas (`start_time`), končni čas (`end_time`), vzorci (`samples`), časovne oznake vzorcev (`times`), časovni interval med vzorci (`sample_time_interval`) in dodane anotacije (`annotations`). Določili smo, da ima posamezen dokument največ 2000 vzorcev, vrstni red dokumentov pa določa zaporedno število dokumenta. Na tak način lahko shranimo poljubno dolg zapis s poljubnim številom odvodov. Najdaljši zapisi, ki smo jih uporabili, so dolgi približno 30 minut.

Primer dokumenta v tem indeksu je na odseku kode 4.5.

```
{
    "record_id": "74404",
    "record_number": 1,
    "lead": "aVR",
    "number_of_samples": 2000,
    "start_time": 4,
    "end_time": 8,
    "samples": [
        -1620,
        -1692,
        ... ],
    "times": null,
    "sample_time_interval": 0.002
}
```

Odsek kode 4.5: Primer dokumenta iz indeksa `record_raw_data`

4.3.3 Indeks z anotacijami hitrosti srčnega utripa

Srčni utrip izračunamo iz RR intervala, ki je časovni interval med dvema sosednjima QRS kompleksoma. Potrebno je anotirati hitrosti srčnega utripa po celem zapisu. [16]

Indeks z anotacijami hitrosti srčnega utripa smo poimenovali `annotations_heart_rate`. Polja, ki se lahko pojavijo v tem dokumentu, so: ID posnetka (`record_id`), zaporedno število dokumenta (`record_number`), začetki intervalov (`start_indexes`) in vrednosti (`values`).

Primer dokumenta v tem indeksu je na odseku kode 4.6.

```
{
    "record_id": "74404",
    "record_number": 0,
    "start_indexes": [
        0,
        1004,
        1024,
        ...
    ],
    "values": [
        91,
        92,
        86,
        ...
    ]
}
```

Odsek kode 4.6: Primer dokumenta iz indeksa `annotations_heart_rate`

4.3.4 Indeks z anotacijami oznak srčnih utripov

Prav tako je pomembna oblika vsakega utripa in kombinacije oblik istega utripa v različnih odvodih ob istem času. Gledata se višina in širina vsakega posameznega dela utripa (P val, QRS kompleks, T val ...). Takšne informacije

se ponavadi shranijo že ob snemanju in so v obliki označbe vrste utripa in kje na posnetku se pojavi. [16]

Ta indeks smo poimenovali `annotations_beat`. Polja, ki se lahko pojavijo v tem dokumentu, so: ID posnetka (`record_id`), zaporedno število dokumenta (`record_number`) in anotacije (`anns`), ki so same poddokument in vsaka vsebuje polja lokacija vzorca (`sample_no`), koda anotacije (`ann_code`) in opis anotacije (`ann_description`).

Primer dokumenta v tem indeksu je na odseku kode 4.7.


```
{
    "record_id": "62652",
    "record_number": 0,
    "anns": [
        {
            "sample_no": 167,
            "ann_code": "N",
            "ann_description": "Normal beat"
        },
        {
            "sample_no": 404,
            "ann_code": "N",
            "ann_description": "Normal beat"
        },
        {
            "sample_no": 637,
            "ann_code": "N",
            "ann_description": "Normal beat"
        },
        {
            "sample_no": 873,
            "ann_code": "N",
            "ann_description": "Normal beat"
        },
        {
            "sample_no": 1108,
            "ann_code": "V",
```

```
            "ann_description": "Premature ventricular
                ↪ contraction"
        }, ...
    ]
}
```

Odsek kode 4.7: Primer dokumenta iz indeksa `annotations_beat`

4.4 Pridobivanje podatkov

Za svojo testno bazo smo uporabili 2 SCP-ECG zapisa in 4 DICOM-ECG zapise, ki smo jih našli na spletu. Iz baze PhysioBank smo uporabili še 104 WFDB zapise.

Anotacije za hitrost srčnega utripa smo dodali 43-im zapisom, anotacije za oznake srčnih utripov pa 28-im zapisom.

Algoritma za pretvorbo binarnih SCP in DICOM datotek smo našli prosto dostopna na GitHub-u [4, 14], za WFDB datoteke pa smo uporabili programsko opremo iz PhysioToolkit-a (ukaza `rdsamp` in `rdann`). Za anotacije hitrosti srčnega utripa smo uporabili algoritem iz GitHub-a [14].

Veliko podatkov, predvsem iz vira PhysioNet, je bilo očiščenih vseh drugih podatkov razen teh o signalu samem. Prav tako so tudi pri SCP-ECG in DICOM-ECG datotekah velikokrat manjkali demografski podatki o pacientih in o snemanju, ali pa algoritem ni omogočal pridobitve vseh informacij iz formata. Zato smo se odločili, da bomo za potrebe ponazoritve delovanja te baze signalom dodali naključno generirane izmišljene podatke o spolu in starosti pacienta ter datumih snemanja.

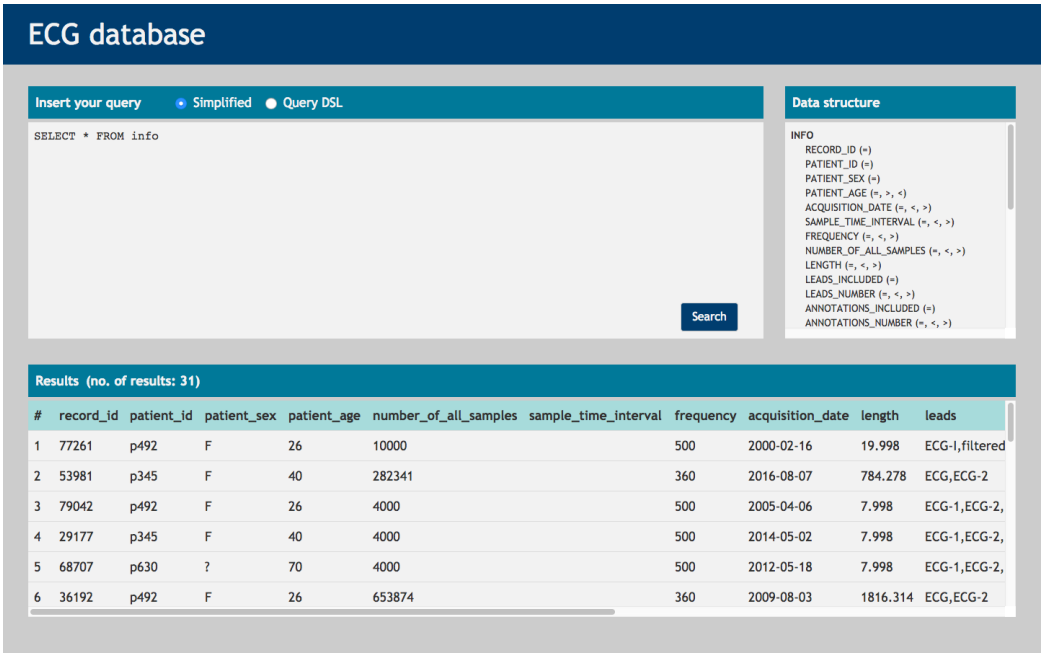
Skupno imamo v bazi tako 110 dokumentov v indeksu `record_info_data`, 23870 dokumentov v indeksu `record_raw_data`, 362 dokumentov v indeksu `annotations_heart_rate` ter 2492 dokumentov v indeksu `annotations_beat`. Skupno zavzemajo približno 1.3 GB prostora.

4.5 Opis uporabniškega vmesnika

Za komunikacijo s podatkovno bazo smo implementirali preprost uporabniški vmesnik, ki ga sestavljajo tri komponente:

- **razdelek za pisanje poizvedbe**, kjer uporabnik napiše svojo poizvedbo v specifičnem poizvedovalnem jeziku, ki smo ga razvili za uporabo naše baze (opisan v nadaljevanju),
- **razdelek z opisom podatkovne strukture**, namenjen lažjemu pisanju poizvedb in
- **razdelek s tabelo rezultatov**, kjer se ob pritisku na gumb za iskanje prikažejo vsi rezultati poizvedbe.

Uporabniški vmesnik je prikazan na sliki 4.1.



Slika 4.1: Poizvedba v uporabniškem vmesniku in rezultat

4.5.1 Implementacija poizvedovalnega jezika

Predpostavili smo, da bi bil tipičen uporabnik naše baze nek strokovnjak iz medicinskega področja, ki ne bi bil tehnično usposobljen za delo z DSL jezikom, ki ga ponuja ElasticSearch. Zato smo v namen dovolj preprostega, a vseeno dokaj fleksibilnega načina iskanja s pomočjo orodja Antlr [1] razvili svoj poizvedovalni jezik in ga implementirali v uporabniški vmesnik.

Antlr (oz. ANother Tool for Language Recognition) je orodje, namenjeno tvorjenju razčlenjevalnikov (ang. *parser generator*) za branje, procesiranje, izvajanje in prevajanje strukturiranih tekstovnih ali binarnih datotek. Pogosto se ga uporablja za gradnjo programskih jezikov. Kot vhod mu podamo slovnico, na podlagi katere generira razčlenjevalnik, ki lahko gradi in se sprehaja po razčlenitvenih drevesih. [1]

Tudi za svoj poizvedovalni jezik smo zasnovali slovnico - prikazana je na sliki 4.2. Na podlagi te smo kreirali razčlenjevalnik in spehajalca po abstraktnem sintaktičnem drevesu, ki iz našega poizvedovalnega jezika avtomatsko tvorita poizvedbe za ES bazo.

Poizvedovalni jezik smo razvili v smislu zelo poenostavljenega jezika SQL. Vsaka poizvedba ima tri glavne dele: del SELECT, del FROM in del WHERE, kjer prvi opiše, katera polja nas zanimajo, drugi iz katerega indeksa poizvedujemo in tretji katerim pogojem morajo ustrezati iskani dokumenti. Primer SELECT poizvedbe je na odseku kode 4.8, kjer želimo iz indeksa info (`record_info_data`) dobiti seznam vseh zapisov, ki so daljši od 60 sekund, zanimajo pa nas polja ID zapisa (`record_id`), spol pacienta (`patient_sex`) in pa starost pacienta (`patient_age`).

```
SELECT record_id, patient_sex, patient_age
FROM info
WHERE length > 60
```

Odsek kode 4.8: Primer SELECT poizvedbe

Ker nam tehnologija ElasticSearch ne omogoča stikov med indeksi, smo za takšne poizvedbe, ki bi to zahtevale, sestavili poizvedovanje v več kora-

```
grammar Search;

query                : ( select_query | get_query )+ EOF ;
select_query         : select from where? ;
get_query            : get from where? ;
select               : SELECT elements ;
from                 : FROM sources ;
where                : WHERE conditions ;
get                  : GET data_type data_name ;
data_type            : DATA_LIST | RECORD_LIST ;
data_name            : WORD ;
elements             : ( field )+ ;
sources              : ( WORD )+ ;
conditions           : condition | conditions AND conditions | conditions OR conditions ;
condition            : term operator value ;
term                 : field ;
field                : WORD | ( WORD '.' WORD) | '*' ;
operator             : '=' | '<' | '>' ;
value                : ( decimal | date | VARIABLE | CODE | NUMBER | WORD | ID ) ;
date                 : NUMBER ( '.' | '-' ) NUMBER ( '.' | '-' ) NUMBER ;
decimal              : ( DIGIT | NUMBER ) '.' ( DIGIT | NUMBER ) ;
```

Slika 4.2: Del slovnice za poizvedovalni jezik v Antlr-ju

kih. Glavni polji za povezovanje med indeksi sta v našem primeru id zapisa (`record_id`) in zaporedno število dokumenta (`record_number`). Zato smo razvili možnost delnih poizvedb, ki nam vračajo seznam ID-jev ali seznam zaporednih števil dokumentov v oblikah, primernih za nadaljnje poizvedovanje. Delne poizvedbe imajo strukturo GET-FROM-WHERE, kjer v delu GET specificiramo kakšen rezultat vračamo (`record_list` za seznam ID-jev zapisov in `data_list` za seznam zaporednih števil dokumentov) in ime spremenljivke, v katero naj se shrani. Primer GET poizvedbe je na odseku kode 4.9, kjer v spremenljivko `rl` shranimo vse ID-je zapisov, ki ustrezajo danim pogojem.

```
GET record_list rl
FROM info
WHERE length > 60
```

Odsek kode 4.9: Primer GET poizvedbe

Za lažje pisanje poizvedb smo v razdelku z opisom podatkovne strukture navedli vse indekse in polja, po katerih lahko iščemo ter kakšne operatorje lahko uporabljamo. V tabeli rezultatov pa so prikazani vsi rezultati naše poizvedbe.

Poglavje 5

Evalvacija rešitve

V tem poglavju bomo najprej evalvirali našo rešitev izdelave EKG aplikacije prek izvedbe petih uporabniških scenarijev, pri čemer bomo opazovali, koliko poizvedb moramo sestaviti za željen rezultat (oz. v kolikšni meri sploh uspemo izvesti zahtevano iskanje), kompleksnost in trajanje poizvedb. Ponazorili bomo iskanje prek uporabniškega vmesnika naše aplikacije in prek orodja, ki ga ponuja PhysioNet (PhysioBank Search Record) in ju primerjali.

V naslednjem podpoglavju pa si bomo ogledali še uspešnost kompleksnejšega poizvedovanja po obeh bazah brez uporabniškega vmesnika in tako evalvirali našo rešitev še iz semantičnega vidika, pri čemer sta naša glavna kriterija, kako kompleksne poizvedbe je mogoče izvesti in kako tehnično usposobljen mora biti uporabnik, da jih lahko sestavi.

5.1 Testna konfiguracija

Uporabniške scenarije smo izvajali na virtualni napravi v okolju VMware Workstation 12 z naslednjimi specifikacijami:

- procesor: 4.00 GHz Intel Core i7-4790K,
- pomnilnik: 3.80 GB,
- operacijski sistem: Ubuntu 16.04 LTS, 64-bit.

Uporabili smo ElasticSearch verzijo 6.2.2 na lokalnem strežniku. Večino nastavitev za delovanje strežnika in gruče smo pustili privzete, kot je opisano v dokumentaciji [7]. Vsak indeks je razdeljen na 5 črepinj, vsaka črepinja ima eno kopijo.

Potrebno je poudariti, da je z iskanjem po bazi PhysioNet mišljeno samo iskanje po indeksni datoteki s programsko opremo pbsearch [11]. Za merjenje trajanja poizvedb smo indeksno datoteko priredili v velikost naše baze (vsebuje 110 posnetkov). Slike iskanja smo sicer posneli na spletnem iskalniku PhysioBank Record Search (ki vsebuje veliko več posnetkov) v namen lažje predstavitve, kakšno je iskanje za zdravnike.

Čase trajanja poizvedb v obeh bazah smo pridobili tako, da smo 10x izmerili čas trajanja posamične poizvedbe in izračunali povprečje. Vključili smo jih bolj za pridobitev občutka o trajanju poizvedb, saj je tako različni tehnologiji (iskanje po celi bazi v primerjavi z iskanjem po eni sami datoteki) težko primerjati samo na podlagi časov trajanj poizvedb.

Omenili pa bi še, da ima uporabljen ES strežnik na voljo le minimalne zahteve, kar se tiče strojne opreme, saj bi za svoje optimalno delovanje potreboval več kot le eno vozlišče. Iz tega razloga so izmerjeni časi poizvedovanja precej veliki, a ker ti niso ključnega pomena za naš raziskovalni problem, takšna konfiguracija zadostuje.

5.2 Evalvacija z uporabniškimi scenariji

Uporabniške scenarije smo si zamislili na podlagi tipičnih primerov iskanja navedenih v PhysioNet viru [12]. Predpostavljamo, da bi zdravniki in raziskovalci želeli iskati predvsem po demografskih podatkih o pacientih, po določenih diagnozah, vsebovanih odvodih, po določenih časovnih odsekih znotraj posnetkov, specifičnih anotacijah in tako dalje. Zamislili smo si 5 primerov iskanja, ki so v naslednjih podpoglavjih bolj podrobneje opisani in obenem tudi predstavimo, kako jih izvedemo s pomočjo našega uporabniškega vmesnika.

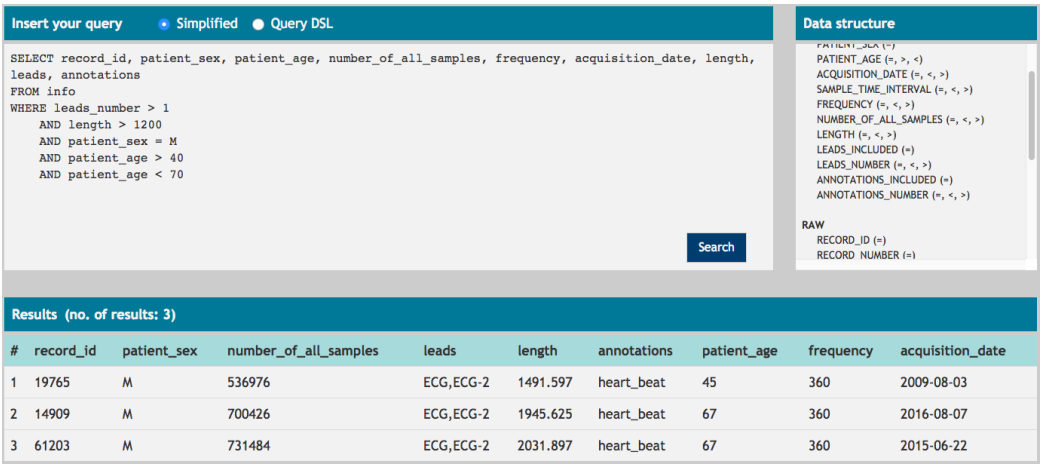
5.2.1 Vsi zapisi, ki vsebujejo dva ali več odvodov, so vsaj 20 minut dolgi in pripadajo moškim pacientom med starostjo 40 in 70 let

Naša rešitev

Vsi zahtevani parametri (število odvodov, dolžina posnetka, spol in starost pacienta) se nahajajo v metapodatkih, zato potrebujemo samo eno poizvedbo nad indeksom `info` (`record_info_data`) (odsek kode 5.1 oz. slika 5.1).

```
SELECT record_id, patient_sex, patient_age,
    ↪ number_of_all_samples, frequency, acquisition_date,
    ↪ length, leads, annotations
FROM info
WHERE leads_number > 1
    AND length > 1200
    AND patient_sex = M
    AND patient_age > 40
    AND patient_age < 70
```

Odsek kode 5.1: Poizvedba v poenostavljeni sintaksi

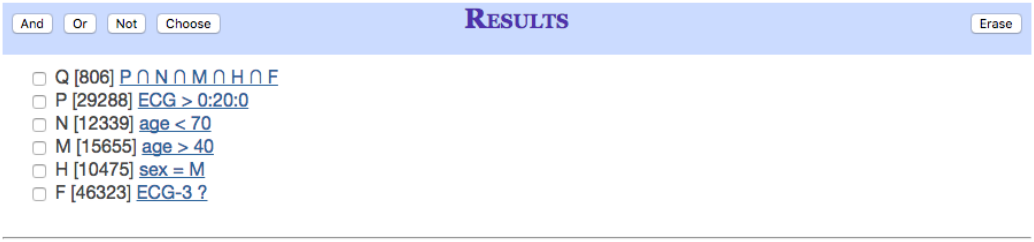


Slika 5.1: Poizvedba v uporabniškem vmesniku in rezultat

Čas trajanja poizvedbe v bazi: 1541 ms.

PhysioNet

Za reševanje te naloge uporabimo orodje PhysioBank Record Search. To omogoča iskanje po le enem polju naenkrat, zato moramo za vsak kriterij izvesti svojo poizvedbo. Vsaka poizvedba vrača seznam ID-jev zapisov. Na koncu naredimo še presek vseh dobljenih seznamov z operatorjem "ali" (*Or*) in tako dobimo končen seznam ID-jev. Poizvedbe so prikazane na sliki 5.2.



Slika 5.2: Poizvedba v uporabniškem vmesniku in rezultat

Seštevek trajanja posamičnih poizvedb v bazi: 107.7 ms.

Primerjava

V obeh primerih uspemo sestaviti pravo poizvedbo in dobiti željene rezultate.

Na platformi PhysioNet moramo zaradi možnosti iskanja po samo enem polju naenkrat sicer izvesti 5 posamičnih poizvedb, preden pridemo do rezultata, z ElasticSearch rešitvijo pa samo eno.

Prednost ElasticSearch rešitve je tudi v tem, da lahko izbiramo, katere izmed metapodatkov zapisa želimo še prikazati, pri PhysioNet-u pa dobimo samo ID-je zapisov.

PhysioNet pa ima sicer drugo prednost, kar je, da lahko zaradi gradnje preprostih poizvedb in možnosti združevanja le teh s pomočjo operatorjev dobimo nove poizvedbe, ki jih potem lahko še naprej združujemo in na ta način sestavimo veliko bolj kompleksno poizvedbo, kakor jo omogoča naš vmesnik za ElasticSearch bazo.

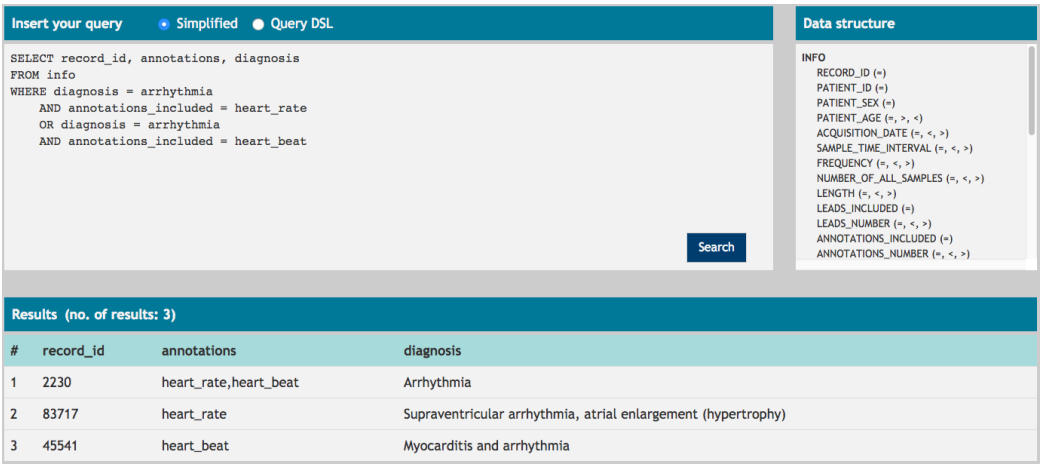
5.2.2 Vsi zapisi z diagnozo aritmije, ki tudi vsebujejo anotacije o hitrosti ali oznakah srčnega utripa

Naša rešitev

Tudi tokrat se vsi zahtevani parametri (diagnoza in prisotne anotacije) nahajajo v metapodatkih, zato potrebujemo samo eno poizvedbo nad indeksom `info` (`record_info_data`) (odsek kode 5.2 oz. slika 5.3). Pri tej poizvedbi lahko izkoristimo možnost iskanja po celotnem besedilu nad poljem diagnoza, zaradi česar našemu iskanju ustrezajo npr. tudi dokumenti z diagnozo "Supraventricular arrhythmia" (kot je prikazano na sliki 5.3) in tako izboljšamo iskanje.

```
SELECT record_id, annotations, diagnosis
FROM info
WHERE diagnosis = arrhythmia
    AND annotations_included = heart_rate
    OR diagnosis = arrhythmia
    AND annotations_included = heart_beat
```

Odsek kode 5.2: Poizvedba v poenostavljeni sintaksi



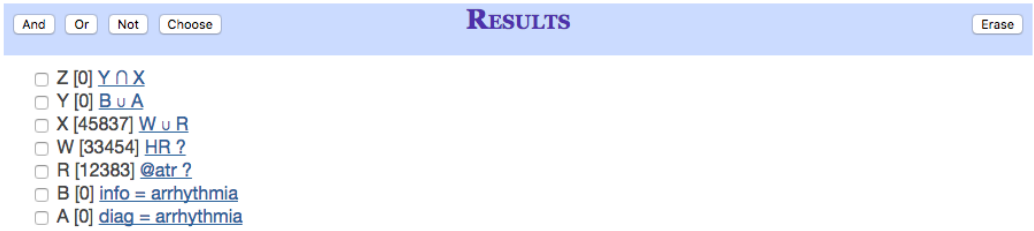
Slika 5.3: Poizvedba v uporabniškem vmesniku in rezultat

Čas trajanja poizvedbe v bazi: 628.3 ms.

PhysioNet

Tudi pri tej nalogi uporabimo orodje PhysioBank Record Search. Uspešno najdemo zapise, ki vsebujejo zahtevane anotacije, ne najdemo pa nobenega, ki bi vseboval diagnozo aritmije (iščemo po poljih `diagnosis` in `info`, v katerih naj bi se nahajale diagnoze). Ob ročnem pregledu baze ugotovimo, da takšni zapisi vseeno obstajajo. Eden izmed njih ima diagnozo "There is borderline first degree AV block and sinus arrhythmia.", vendar ga orodje PhysioBank Record Search ne najde, saj zahteva, da je tekstovno polje identično našemu iskalnemu kriteriju, ki je sicer samo "arrhythmia". Izvedene poizvedbe so prikazane na sliki 5.4.

Seštevek trajanja posamičnih poizvedb v bazi: 69.7 ms.



Slika 5.4: Poizvedba v uporabniškem vmesniku in rezultat

Primerjava

V obeh primerih uspemo sestaviti želeno poizvedbo. Pri tem primeru opazimo še eno prednost uporabe tehnologije ElasticSearch, kar je možnost iskanja po celotnem besedilu po poljih. Ta možnost je zelo koristna v primerih, če želimo iskati po diagnozah ali komentarjih, ki so dodani zapisom s strani zdravnikov in so ponavadi neka daljša besedila.

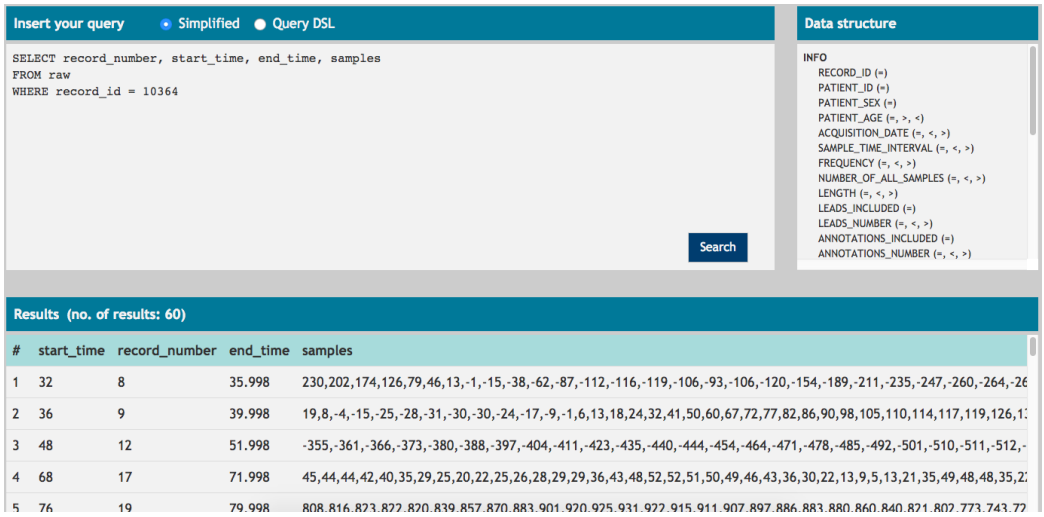
5.2.3 Celoten signal iz zapisa z danim ID-jem zapisa

Naša rešitev

Izberemo si zapis z ID-jem 10364. Tokrat poizvedbo izvedemo nad indeksom `raw` (`record_raw_data`), kjer dobimo vse dokumente, ki vsebujejo informacijo o signalu od zapisa z ID-jem 10364. Poizvedba je prikazana na odseku kode 5.3 in na sliki 5.5

```
SELECT record_number, start_time, end_time, samples
FROM raw
WHERE record_id = 10364
```

Odsek kode 5.3: Poizvedba v poenostavljeni sintaksi



Slika 5.5: Poizvedba v uporabniškem vmesniku in rezultat

Čas trajanja poizvedbe v bazi: 678.1 ms

PhysioNet

Te poizvedbe prek orodja PhysioBank Record Search ne moremo izvesti, ker nam le-to vrača samo ID-je zapisov, ne pa tudi njihovih signalov. Grafičen prikaz zapisa lahko prikažemo s pomočjo orodij LightWAVE ali PhysioBank

ATM, kjer zapis najdemo prek njegovega ID-ja, sicer pa lahko primerno datoteko ročno poiščemo v bazi v binarni obliki in ga pretvorimo s pomočjo programske opreme PhysioToolkit.

Primerjava

Z uporabo naše rešitve uspemo priti do željenih rezultatov z eno samo poizvedbo. Na platformi PhysioNet na nek način prav tako, saj samo odpremo datoteko, katere ime je kriterij, po katerem iščemo. Moramo jo sicer še ročno pretvoriti iz binarne v tekstovno obliko.

5.2.4 Del zapisa na intervalu med 10 in 20 sekundami s pripadajočimi anotacijami

Naša rešitev

Tokrat moramo za pridobitev željenih rezultatov izvesti več delnih poizvedb. Izberemo zapis z ID-jem 42205.

Anotacije so prek polja `record_number` vezane na posamezne datoteke iz indeksa `raw` (`record_raw_data`) (število datotek v indeksu `raw`, ki spadajo k določenemu zapisu, je enako številu datotek v indeksu katerih koli anotacij, ki spadajo k istemu zapisu). Zato moramo najprej iz indeksa `raw` pridobiti seznam primernih kosov signala in prek tega bomo dobili anotacije, ki spadajo v dan časovni okvir.

Najprej izvedemo poizvedbo GET nad indeksom `raw`, kjer dobimo vsa zaporedna števila dokumentov, ki pripadajo zapisu z ID-jem 42205 in vsebujejo signal v času od 10 do 20 sekund od začetka snemanja, in jih shranimo v spremenljivko `dl`.

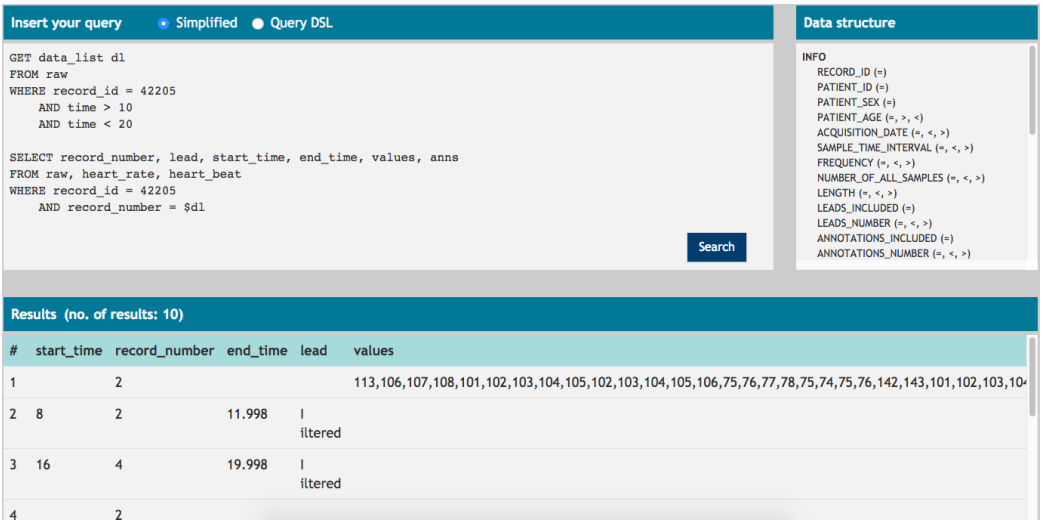
Nato pa izvedemo še poizvedbo SELECT, s katero pridobimo vse dokumente, ki opisujejo izvorne podatke ali dane anotacije. Ustrezati morajo danemu ID-ju zapisa in zaporednim številom dokumentov, danim v spremenljivki `dl`.

Poizvedbi sta prikazani na odseku kode 5.4 in na sliki 5.6.

```
GET data_list dl
FROM raw
WHERE record_id = 42205
    AND time > 10
    AND time < 20

SELECT record_number, lead, start_time, end_time, values,
   ↪ anns
FROM raw, heart_rate, heart_beat
WHERE record_id = 42205
    AND record_number = $dl
```

Odsek kode 5.4: Poizvedba v poenostavljeni sintaksi



Slika 5.6: Poizvedba v uporabniškem vmesniku in rezultat

Seštevek časov trajanj vseh poizvedb v bazi: 922.8 ms.

PhysioNet

Takšne poizvedbe baza PhysioNet ne omogoča, saj ne omogoča iskanja znotraj posameznega zapisa.

Primerjava

Z našo rešitvijo pridemo do željenih rezultatov, vendar moramo izvesti dve poizvedbi. V drugi poizvedbi izkoristimo možnost tehnologije ElasticSearch, da iščemo po treh indeksih naenkrat, zaradi česar nam ni treba izvajati treh posamičnih poizvedb na vsak indeks posebej.

V bazi PhysioNet takšne poizvedbe ne moremo izvesti. Pri tem primeru naletimo na veliko slabost PhysioNet baze, kar je pomanjkanje možnosti iskanja znotraj posamičnih datotek, za kar je kriva že sama narava baze kot zbirke binarnih datotek.

5.2.5 Deli vseh zapisov od določenega pacienta, kjer je hitrost srčnega utripa večja od 80 utripov na minuto

Naša rešitev

Te naloge se, tako kot prejšnje, lotimo po korakih. Izberemo pacienta z ID-jem p345.

Začnemo s poizvedbo GET, s katero izmed vseh zapisov najdemo takšne, ki pripadajo pacientu p345 in jih shranimo v spremenljivko `rl`.

Potem ponovno izvedemo poizvedbo GET, s katero najdemo vsa zaporedna števila dokumentov, ki spadajo k danim ID-jem zapisov (oz. danemu pacientu) in vsebujejo vrednosti srčnega utripa nad 80 utripov na minuto. Te shranimo v spremenljivko `dl`.

Na koncu izvedemo še poizvedbo nad indeksom `raw` (`record_raw_data`), kjer dobimo vse dele signala, ki spadajo k danim ID-jem (spremenljivka `rl`) in danim zaporednim številom dokumentov (spremenljivka `dl`).

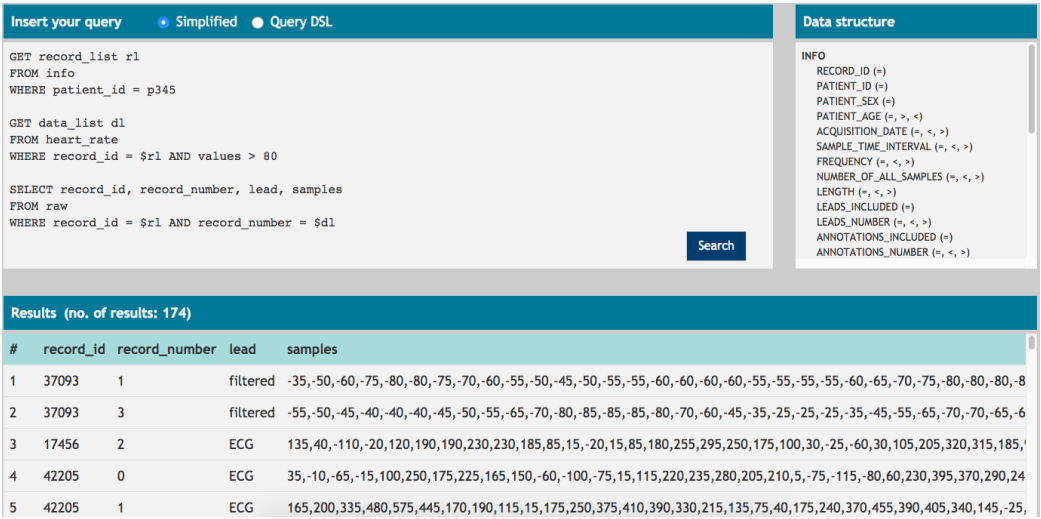
Poizvedba je prikazana na odseku kode 5.5 in na sliki 5.7.

```
GET record_list rl
FROM info
WHERE patient_id = p345
```

```
GET data_list dl
FROM heart_rate
WHERE record_id = $rl AND values > 80

SELECT record_id, record_number, lead, samples
FROM raw
WHERE record_id = $rl AND record_number = $dl
```

Odsek kode 5.5: Poizvedba v poenostavljeni sintaksi



Slika 5.7: Poizvedba v uporabniškem vmesniku in rezultat

Seštevek časov trajanj vseh poizvedb v bazi: 2161 ms.

PhysioNet

Takšne poizvedbe baza PhysioNet ne omogoča. Lahko poiščemo vse zapise, ki pripadajo danemu pacientu in vsebujejo anotacije o srčnem utripu, vendar ne moremo iskati po vrednostih srčnega utripa znotraj določenega zapisa.

Primerjava

Naša rešitev omogoča takšno poizvedbo, vendar na tem primeru vidimo, da čim želimo izvesti iskanje na več indeksih, se pisanje poizvedbe hitro zaplete

in posledično je treba izvesti večje število poizvedb na bazo.

Če iskanje na posameznih indeksih ni odvisno od rezultatov predhodnih iskanj na drugih indeksih, poizvedbe lahko izvedemo vzporedno, kar nam omogoča tehnologija ES (kot smo prikazali v prejšnjem primeru). Čim pa je poizvedovanje serijsko in so naslednje poizvedbe odvisne od prejšnjih, se zaradi narave nerelacijske baze in pomanjkanja možnosti delanja stikov med indeksi poizvedba zaplete in razvleče.

Čeprav se z našo rešitvijo poizvedba nekoliko zakomplicira, se po drugi strani na vrsti baze, kot jo uporabljajo pri PhysioNet-u, te poizvedbe sploh ne da izvesti oz. jo lahko izvedemo le do mere, ko še poizvedujemo med zapisi in ne znotraj enega zapisa.

5.3 Naprednejše iskanje

Predpostavimo, da je naš uporabnik bolj tehnološko podkovan. Pogledali si bomo, kako bi z iskanjem neposredno po podatkovni bazi izvedel naslednji uporabniški scenarij.

5.3.1 Ime in trajanje prvih treh najdaljših zapisov, ki vsebujejo vsaj 3 odvode, urejenih po dolžini zapisa

Naša rešitev

Potrebno je izvesti eno poizvedbo v indeksu `record_info_data`. Prikazana je na odseku kode 5.6. Podatkovno bazo naslovimo prek REST API-ja na vratih 9200.

Ključ `size` določa željeno število vrnjenih rezultatov, ključ `_source` določa polja, ki jih želimo, ključ `sort` določa pravila za urejanje po vrsti, ključ `query` pa pravila za filtriranje rezultatov.

Poizvedba je na odseku kode 5.6, rezultat pa na odseku kode 5.7, kjer so naši rezultati navedeni v polju s ključem `_source`.

```
curl -XGET 'localhost:9200/record_info_data/_search?pret
ty' -H 'Content-Type: application/json' -d'
{
    "size": 3,
    "_source": ["record_id", "length"],
    "sort": { "length" : {"order" : "desc"}},
    "query": {
        "script": {
            "script": "doc.leads.length > 3"
        }
    }
}
'
```

Odsek kode 5.6: Poizvedba v domensko specifičnem jeziku za ElasticSearch

```
{
  "took" : 329,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 13,
    "max_score" : null,
    "hits" : [
      {
        "_index" : "record_info_data",
        "_type" : "recs",
        "_id" : "Jv5dwmEBOIPBSIJDEgE0",
        "_score" : null,
        "_source" : {
          "record_id" : "88145",
          "length" : 123.1
        },
```

```
        "sort" : [
          123.1
        ]
      },
      {
        "_index" : "record_info_data",
        "_type" : "recs",
        "_id" : "wP5dwmEBOIPBSIJDWQJW",
        "_score" : null,
        "_source" : {
          "record_id" : "19723",
          "length" : 114.596
        },
        "sort" : [
          114.596
        ]
      },
      {
        "_index" : "record_info_data",
        "_type" : "recs",
        "_id" : "rv0swmEBOIPBSIJDmOC1",
        "_score" : null,
        "_source" : {
          "record_id" : "63248",
          "length" : 10
        },
        "sort" : [
          10.0
        ]
      }
    ]
  }
}
```


Odsek kode 5.7: Rezultat poizvedbe na odseku kode 5.6

PhysioNet

Rešitev, ki jo za takšno iskanje predlagajo na platformi PhysioNet, je uporaba operacij ukazne lupine, ki jih omogočajo Unix operacijski sistemi, nad indeksno datoteko.

Najprej z ukazom `grep` poiščemo vse takšne zapise, ki vsebujejo odvod ECG3, torej morajo vsebovati vsaj 3 odvode (odsek kode 5.8).

```
grep ECG3 physiobank-index
```

Odsek kode 5.8: Ukaz grep

Nato z ukazom `cut` izločimo vsa polja, razen tistih z imenom in dolžino zapisov (odsek kode 5.9).

```
cut -f 1,6
```

Odsek kode 5.9: Ukaz cut

Z ukazom `sort` zapise še uredimo po padajočih vrednostih drugega polja in z ukazom `head` prikažemo samo prve tri (odseka kode 5.10 in 5.11).

```
sort -nr -k2
```

Odsek kode 5.10: Ukaz sort

```
head -3
```

Odsek kode 5.11: Ukaz head

Celoten ukaz, zložen v cevovod, je prikazan na odseku kode 5.12.

```
grep ECG3 physiobank-index | cut -f 1,6 |
sort -nr -k2 | head -3
```

Odsek kode 5.12: Celotna poizvedba na indeksni datoteki

Rezultati so prikazani na odseku kode 5.13.

```
mimic2wdb/31/3101148/   2299827
mimic2wdb/37/3752730/   1845755
mimic2wdb/32/3212213/   1842540
```

Odsek kode 5.13: Rezultat poizvedbe na indeksni datoteki

Primerjava

V obeh primerih mora uporabnik poznati neko bolj specifično tehnologijo. Pri uporabi ES mora znati pošiljati in prejemati podatke prek protokola HTTP, poleg tega pa se mora naučiti domensko specifičen jezik (DSL) za ElasticSearch, ki se ga uporablja izključno pri tej tehnologiji.

Pri iskanju po indeksni datoteki mora po drugi strani poznati vsaj osnovne ukaze za delo z besedilom v ukazni lupini. Glede na to, da je ta skupna vsem Unix sistemom, lahko trdimo, da je skupnost uporabnikov večja in bi zaradi tega hitreje našel podporo pri uporabi.

Če primerjamo sami poizvedbi, že na prvi pogled vidimo, kako preprostejša in bolj jedrnata je poizvedba z uporabo ukazne lupine. Vendar pa ima ta način poizvedovanja svoje omejitve. V zgornji poizvedbi smo npr. število odvodov določili na podlagi predpostavke, da gredo poimenovanja odvodov po zaporedni številki (ECG1, ECG2, ECG3,...), kar ni vedno res. Sešteti vse različne odvode ne bi bilo mogoče. Poleg tega ukazna lupina ne omogoča naprednejših načinov iskanja, kot je npr. iskanje po celotnem besedilu.

S tem primerom smo želeli prikazati, da je uporaba indeksne datoteke kot osnove za iskanje sicer učinkovit način in tudi bolj preprost, kot če uporabljamo npr. ES bazo, a le če je iskanje enostavno. Če pa želimo izvesti kompleksnejše poizvedbe, ta način ne ustreza več.

Poglavje 6

Zaključek

V delu smo najprej preučili današnje stanje na področju digitalne elektrokardiografije in identificirali nekatere težave, ki se trenutno pojavljajo.

S problemom iskanja po bazi EKG zapisov so se delno ukvarjali že raziskovalci na MIT, ki so razvili platformo PhysioNet z bogato bazo EKG podatkov. Njihovo bazo predstavlja velika zbirka binarnih datotek, ki ni bila zasnovana z namenom zapletenejšega iskanja po njej - to funkcionalnost so razvili naknadno.

Naša naloga je bila zasnovati bazo, ki je v prvi vrsti namenjena iskanju, zato smo se odločili za uporabo tehnologije ElasticSearch. Zasnovali smo podatkovno bazo, specializirano za shranjevanje EKG podatkov in njihovih anotacij. V namen preprostega poizvedovanja smo razvili tudi poenostavljen poizvedovalni jezik in uporabniški vmesnik, preko katerega lahko pošiljamo poizvedbe na bazo.

Svojo rešitev smo evalvirali na podlagi šestih uporabniških scenarijev. Glavne prednosti uporabe ES kot podatkovne baze v primerjavi z binarnimi datotekami so možnost iskanja po vseh poljih, ker so vsa indeksirana, ter možnost iskanja tako znotraj posamičnih zapisov kot na nivoju vseh zapisov. Kot smo prikazali v šestem uporabniškem scenariju, nam ES poleg iskanja nudi zelo raznovrstne operacije nad podatki, kot so sortiranje po različnih poljih, filtriranje, izbiranje željenih polj in števila vrnjenih zapisov, agrega-

cije, itd., s čimer se dokaj približa zmožnostim relacijske baze. Edina resnejša slabost, ki smo jo opazili je ta, da se med indeksi ne da delati stikov. ES sicer omogoča poizvedovanje nad več indeksi naenkrat, a če je iskanje po enem indeksu odvisno od rezultatov iskanja po prejšnjem indeksu, smo primorani izvesti več ločenih poizvedb.

Vseeno bi uporabo tehnologije ElasticSearch označili kot učinkovito za shranjevanje in iskanje po EKG podatkih. Predvidevamo, da bi nadaljnje razširitve naše implementacije pokazale še dodatne prednosti. Prva je ta, da je ES zasnovan v namen hranjenja in iskanja po veliki količini podatkov, torej se tudi v primeru, če bi vstavili vse 4 terabajte podatkov, ki jih trenutno vsebuje PhysioNet podatkovna baza, performance iskanja ne bi smele drastično poslabšati. Pri tem predpostavimo, da bazi zagotovimo tudi primerno strojno opremo, saj je delovanje ES osnovano na izdelovanju črepinj in kopij. Druga prednost pa je v tem, da podatkovna shema ni fiksna. Poljubno lahko dodajamo nove indekse z anotacijami in dopolnjujemo stare, kar je pomembno, ker so za učinkovito diagnosticiranje nepravilnosti delovanja srca pogosto pomembne tudi informacije, ki se v samem signalu ne nahajajo ali pa jih moramo naknadno izračunati z algoritmi. Pomembna prednost pa je tudi v tem, da so lahko anotacije v obliki kompleksnejših hierarhičnih oblik, kar zagotavlja JSON format dokumentov.

Zaključili bi z mislijo, da nove tehnologije, kot je tudi ElasticSearch, prinašajo nove možnosti reševanja težav, s katerimi se ukvarja digitalizacija zdravstva, in so zato vsekakor vredne raziskave možnosti uporabe na tem področju.

Literatura

[1] Antlr. http://www.antlr.org/. Dostopano: 20. januar 2018.

[2] Apache Lucene. https://lucene.apache.org/. Dostopano: 20. februar 2018.

[3] Raymond R. Bond, Dewar D. Finlay, Chris D. Nugent, and George Moore. A review of ecg storage formats. *International Journal of Medical Informatics*, 80(10):681 – 697, 2011.

[4] DICOM dekoder. https://github.com/cornerstonejs/dicomParser. Dostopano: 15. december 2017.

[5] Electrocardiography. https://ecg.utah.edu/. Dostopano: 10. december 2017.

[6] ECG stress test. https://www.webmd.com/heart-disease/guide/stress-test. Dostopano: 1. januar 2018.

[7] ElasticSearch documentation. https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html. Dostopano: 10. januar 2018.

[8] Matt B. Oefinger Gari D. Clifford. *Advanced Methods and Tools for ECG Analysis*. Artech House, 2006.

[9] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Mo-

ody, Chung-Kang Peng, and H Eugene Stanley. Current perspective. *Circulation*, 101:e215–e220, 2000.

[10] 24-urno snemanje elektrokardiograma (EKG-ja) ali Holter EKG. `http://www.drmed.org/wp-content/uploads/2014/06/31-Holter-elektrokardiogram.pdf`. Dostopano: 1. januar 2018.

[11] Finding Records in PhysioBank. `https://www.physionet.org/physiobank/database/pbi/#unix-tools`. Dostopano: 14. januar 2018.

[12] PhysioBank Record Search. `https://physionet.org/cgi-bin/pbs/pbsearch`. Dostopano: 4. januar 2018.

[13] PhysioNet. `https://www.physionet.org/about.shtml`. Dostopano: 4. januar 2018.

[14] SCP dekoder, algoritem za detekcijo QRS signalov in HRV analizo. `https://github.com/markozeman/SCP-decoder_QRS-detection_HRV-analysis`. Dostopano: 20. december 2017.

[15] Steven R. Steinhubl and Eric J. Topol. Moving from digitalization to digitization in cardiovascular care: Why is it important, and what could it mean for patients and providers? *Journal of the American College of Cardiology*, 66(13):1489 – 1496, 2015.

[16] Malcolm S. Thaler. *The only EKG book you'll ever need.* LIPPINCOTT WILLIAMS & WILKINS, 7 edition, 2012.

[17] J. D. Trigo, F. Chiarugi, Á. Alesanco, M. Martínez-Espronceda, L. Serrano, C. E. Chronaki, J. Escayola, I. Martínez, and J. García. Interoperability in digital electrocardiography: Harmonization of iso/ieee x73-phd and scp-ecg. *IEEE Transactions on Information Technology in Biomedicine*, 14(6):1303–1317, Nov 2010.