

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Priit Paluoja

**Andmebaas ja selle veebipõhine kasutajaliides loote
geneetilise sõeluuringu analüüsiks**

Bakalaureusetöö (9 EAP)

Juhendajad: Kaarel Krjutškov, PhD

Priit Adler, PhD

Tartu 2017

Andmebaas ja selle veebipõhine kasutajaliides loote geneetilise sõeluuringu analüüsiks

Lühikokkuvõte

Bakalaureusetöö raames valmis uudne **tarkvara** Tervisetehnoloogiate Arenduskeskuse AS (Tervise-TAK) poolt väljatöötamisel oleva tervishoiuteenuse – **geenitesti** jaoks. Teenus sisaldab lapseootel naise ehk **patsiendi** varajast geneetilist sõeluuringut, et tuvastada loote kromosoomi võimalikke väärarenguid.

Loodud tarkvara on multidistsiplinaarne, hõlmates endas patsiendi kliinilise proovi ja isikuandmete teekonda arsti kabinetist laborisse, läbi andmeanalüüsi protsesside kuni automaatse raporti genereerimiseni. Protsess lõpeb arsti teavitamisega, et geenitest on valmis ja tulemused on koondatud raportisse. Andmeanalüüsi algoritmi välja töötamine ei olnud autori ülesanne, kuid autor tegi andmeanalüüsi väljatöötajatega koostööd, et luua koos toimiva lahenduse testversioon.

Tarkvara kiirendab geenitesti tulemuste jõudmist patsiendini, vähendab andmete dubleerimist ja sellest tekkivate sisestusvigade tõenäosust. Lisaks hõlbustab tarkvara geenitesti tellimist, tulemuste verifitseerimist ning on veebipõhise lahendusena tervishoiuteenuse pakkujatele hõlpsalt kasutusele võetav.

Tegu on unikaalse veebirakendusega Eestis, sest töö kirjutamise hetkel tellivad tervishoiuteenuse pakkujad analoogseid geeniteste ilma tarkvaralahenduseta. Eelnevale toetudes tõstab loodud lahendus tervishoiu üldist kvaliteeti.

Võtmesõnad

Veebirakendus, andmebaas, geenitest, NIPT

CERCS

P175: Informaatika, süsteemiteooria

Database with web interface for prenatal genetic screening

Abstract

The main goal of this Bachelor's thesis was to create a software solution for a soon to be available service by the Competence Centre on Health Technologies (CCHT). The service

includes early genetic screening for pregnant women to identify the most common fetal chromosomal anomalies.

The created software solution is multi-disciplinary, covering the entire testing procedure from biological sampling from the doctor's office to automatic report generation. Data analysis software development was not author's responsibility, but required author's cooperation to create a workable test version of the given solution.

The web application accelerates the screening process and reduces the duplication of data and the number of errors caused by this. This solution can be easily introduced to healthcare providers as it is web-based.

It is a unique solution in Estonia as at the time of the writing, no alternative software exists for such fetal screening. On this basis, this solution can heighten the Estonian health care quality.

Keywords

Web application, database, genetic testing, NIPT

CERCS

P175: Informatics, systems theory

Sisukord

1. SISSEJUHATUS.....	6
2. RAKENDUSE ÜLEVAADE.....	8
3. NÕUDED.....	10
3.1 PATSIENDI ÜLEVAATUS	10
3.2 RAPORTI LOOMINE	11
3.3 RAPORTI VAATAMINE.....	12
3.4 KASUTAJATE LISAMINE	12
3.5 ÜLDISED NÕUDED.....	13
4. ARHITEKTUUR.....	14
4.1 VEEBIRAKENDUSE ARHITEKTUUR	14
4.2 LAHENDUSE ARHITEKTUUR TERVIKUNA.....	15
5. INSENER-TEHNILISED VALIKUD.....	17
5.1 VEEBIRAAMISTIK	17
5.2 PORDITAVA DOKUMENDIFORMAADI GENEREERIJ.....	18
5.3 ID-KAART	18
5.4 E-POST	19
5.5 KUJUNDUS.....	19
5.6 VORMI VALIDEERIMINE.....	19
5.7 KASUTAJAÕIGUSED	20
5.8 TÕLGE	21
5.9 ANDMEBAAS	21
5.10 ANDMEMUDEL	22
6. JUHISED.....	25
7. KVALITEET JA NÕUTELE VASTAVUS.....	26
8. EDASIARENDUSVÕIMALUSED	28
9. KOKKUVÕTE	29
AUTOR TÄNAB	31
VIIDATUD KIRJANDUS	32
LISAD	38
I NATERA NIPT TELLIMISE VORM	38
II HARMONY NIPT TELLIMISE VORM	39
III KASUTUSLUGU PATSIENDI ÜLEVAATUSE LISAMISEST.....	40
IV KASUTUSLUGU RAPORTI LOOMISEST.....	41

V KASUTUSLUGU RAPORTI ALLA LAADIMISEST	42
VI KASUTUSLUGU KASUTAJA LISAMISEST	43
VII KASUTUSLUGU PAROOLI MUUTMISEST	44
VIII KASUTUSLUGU ALLUVA ÕIGUSTE MUUTMISEST	45
IX LITSENTS.....	46

1. Sissejuhatus

Loote geneetiline uuring raseduse ajal on tarvilik, et kindlustada terve lapse sünd või tuvastada terviseriske juba varakult, et vanemad saaksid valmistuda haige lapse sünniks. Kuni kümnendikul kõigist geenitestidega kliiniliselt uuritud rasedustest esineb loote genoomis kromosoomi trisoomia või monosoomia [1]. Tegemist on geneetilise vääringuga ehk sündroomiga, mis lõppeb lapse varase surma või terve pika elu kestva vaimse ja füüsilise mahajäämusega. Üheks selliseks vääringuks on näiteks Downi sündroom. Siiani on lapseootel naistel olnud kaks valikut loote geneetiliseks sõeluuringuks [2]. Esimene on **mitteinvasiivne** ema vereproovi analüüs, mille käigus määratakse biomarkerite sisaldust ema veres [3] [4]. Teise, **invasiivse** analüüsi käigus võetakse proov lootevedelikust või platsentast [5].

Seni kasutatud mitteinvasiivsed testid on lootele küll ohutud, kuid need ei ole piisavalt täpsed. On näidatud, et vereproovi põhised testid annavad valenegatiivseid tulemusi 12–23% ulatuses ning valepositiivseid vahemikus 1,9–5,2% [6]. Invasiivsed testid on seevastu kõrge täpsusega (97,5–99,8%) [6], ent nendega kaasneb raseduse katkemise ja nurisünnituse risk [7]. Lisaks põhjustavad invasiivsed testid patsiendile psühholoogilist pinget ja stressi.

Tänapäevane alternatiiv mitteinvasiivsele biomarkerite testimisele on *Non-Invasive Prenatal Testing* (mitteinvasiivne sünnieelne loote DNA uuring) ehk **NIPT**, mis põhineb patsiendi vereproovis leiduva genoomse materjali analüüsil. Patsiendi vereproov sisaldab nii naise kui ka loote geneetilist materjali, mis sekveneeritakse ja analüüsitakse bioinformaatiliste meetoditega [8] [9]. NIPT puhul on tegemist meetodiga, mille positiivse testitulemuse korral tehakse patsiendile täiendav invasiivne analüüs, et kinnitada diagnoosi. Kuna NIPT on erakordselt täpne, siis selle laiem rakendamine vähendaks invasiivse ehk riskidega seotud sõeluuringu vajadust [9].

Tänase seisuga tellitakse NIPT teste Eestis posti teel, sest puudub sobilik tarkvara. NIPT geenitesti pakkujate tellimissüsteeme ei ole autorile teadaolevalt seni kirjeldatud ning iga uus analüüsimeetod vajab oma andmebaasi struktuuri ja tarkvara funktsionaalsust. NIPT testi pakkujate elektroonilise tellimissüsteemi kättesaamatuse tõttu täidab näiteks SA Tartu Ülikooli Kliinikum testi tellimiseks firma Natera (lisa I Natera NIPT tellimise vorm, joonis 6) või Harmony tellimisvormi (lisa II Harmony NIPT tellimise vorm, joonis 7). Paberile andmete kirjutamine ning hiljem teenusepakkuja poolt digitaalseks teisendamine suurendab andmete ümberkirjutamisel tekkiva vea tõenäosust. Seevastu elektrooniline süsteem võimaldab

vereproovi ja andmete saatmise eraldada: andmed saadetakse elektrooniliselt ja vereproov kulleriga. Lisaks saab automatiseerida tulemustest teavitamise ja võimaldada tulemustele ligipääsu kõikjalt, kus on internetiühendus. Veebipõhine vorm võimaldab sisendit kontrollida juba andmete sisestamisel.

Käesoleva bakalaureusetöö raames loodi tarkvara testversioon Tervise-TAK poolt väljatöötamisel oleva NIPT tervishoiu teenuse jaoks. Tarkvara teeb tervishoiuteenusele võimalikuks veebipõhiselt NIPT geenitesti tellimise ja tulemustest teavitamise. Tegu on autorile teadaolevalt Eestis ainulaadse elektroonilise tellimissüsteemiga NIPT geenitesti jaoks.

Bakalaureusetöö koosneb kaheksast peatükist. Teises peatükis antakse ülevaade valminud lahendusest, millele järgneb peatükk nõuetest tarkvarale. Neljandas peatükk esitab arhitektuursed otsused, millele järgnev peatükk keskendub insener-tehnilistele valikutele. Kuuendas peatükis esitatakse juhised rakenduse testimiseks. Seitsmendas peatükis veendutakse rakenduse kvaliteedis. Kaheksandas peatükis tuuakse välja edasiarendusvõimalused ning selgitatakse, miks valmis testversioon. Üheksandas peatükis esitatakse kokkuvõtte. Lisades on toodud kasutuslood ja näidisvormid geenitesti tellimiseks.

2. Rakenduse ülevaade

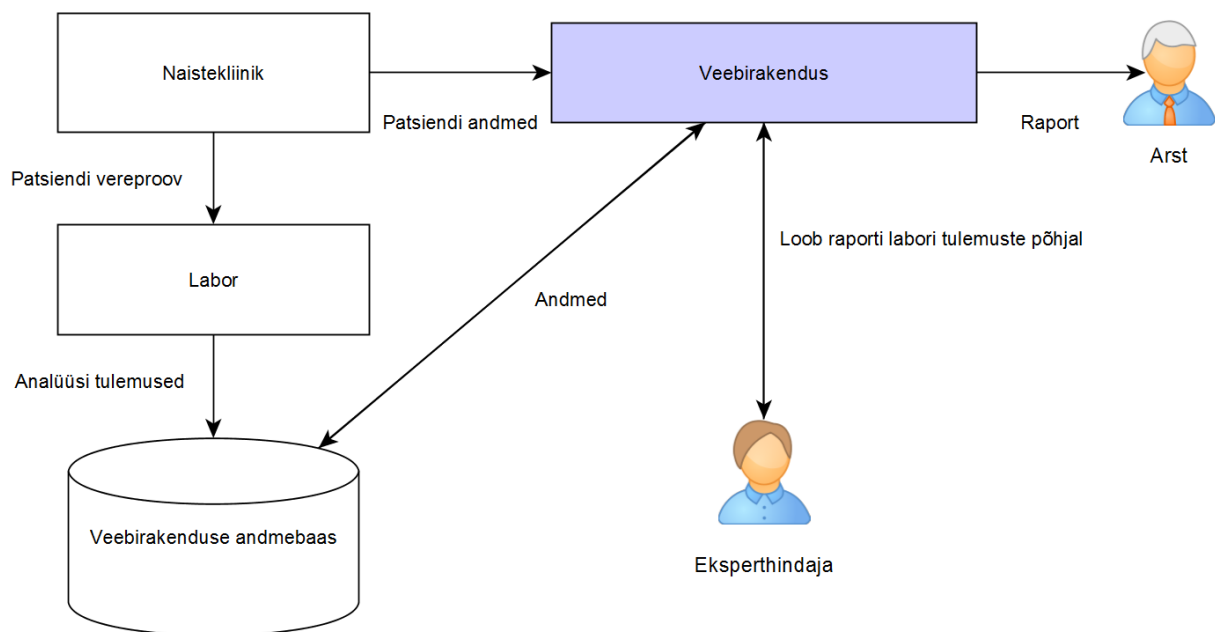
Peatükk annab ülevaate valminud lahendusest.

Käesoleva bakalaureusetöö raames valmis andmebaas ja selle veebiliides Tervise-TAKi poolt peagi pakutava tervishoiuteenuse jaoks. Loodud tarkvara teeb võimalikuks mitteinvasiivse sünnieelse geenitesti (NIPT) elektroonse tellimise, tulemuste hindamise ja protsessis osalejate teavitamise. Bioinformaatilise analüüsi algoritmi osa bakalaureusetöö ei käsitle.

Veebirakendus võimaldab sisestada lapseootel naise testimiseks vajalikud andmed ja kuvada neid hiljem koos analüüsi tarkvara poolt väljastatud andmetega **eksperthindajale** (joonis 1).

Eksperthindaja, kes otsustab andmete põhjal, kas loode on terve või väärarenguga, saab luua veebiliideses raporti, kus kuvatakse kogu patsiendi kohta teadaolev informatsioon, analüüsi tulemused ning kuhu lisandub eksperthindaja otsus (joonis 1).

Pärast eksperthindaja raporti loomist teavitatakse arsti, kelle e-posti aadress sisestati patsiendi andmetega süsteemi, raporti valmimisest. Arst siseneb süsteemi ja otsustab raporti põhjal, kas suunata patsient edasistele uuringutele või mitte (joonis 1).



Joonis 1. Veebirakendus ühendab kliiniku, labori, eksperthindaja ja arsti infovahetuse.

Andmebaasi disaini oli kaasatud nii arste kui ka tulemuste analüüsiga tegelevaid inimesi. Kuna autoril puudus juurdepääs analoogsetele tellimissüsteemidele, disainiti lahendus koostöös Tervise-TAK NIPT geenitesti väljatöötavate inimestega. Tarkvara programmeeris autor iseseisvalt. Valminud lahendus koosneb 66 Java klassist, kus igas klassis on keskmiselt 49

koodirida. Failiridasid Java klassides on kokku 3295, millest 2403 moodustavad koodiread, 278 kommentaarid ja 614 on tühjad read. Projektis on 12 HTML-faili (*HyperText Markup Language*), millest kahte kasutatakse e-posti saatmisel. Lahendus kasutab 13 välist sõltuvust, millest kuus on seotud Spring Boot raamistikuga. Veebiliides toetab eesti, vene ja inglise keelt. Käesoleva töö tulemusel valmis testversioon, milles puuduvad hetkel veel teatud funktsioonid, mis ei võimalda rakendust praktikas kasutada. Üks puuduv funktsionaalsus on logimine.

Valminud lahenduse keerukus seisneb järgnevas:

1. Süsteem töötleb **sensitiivseid terviseandmeid**, mis eeldavad turvalist disaini.
2. Valminud tarkvara on **multidistsiplinaarne**, ühendades geenitesti teostamiseks vajalikud etapid.
3. Lahendus võimaldab geenitesti tulemusi **PDF-kujul** (*Portable Document Format*) alla laadida.
4. Tarkvara kasutab **Eesti ID-kaardil** põhinevat isikutuvastust.
5. Veebiliidese kasutajate **õigused** on täpselt **piiritletud**.
6. Lahenduse veebiliides on **kolmekeelne**.

Valminud tarkvara lähtekood on soovi korral kättesaadav käesoleva töö ülevaatajatele versioonihaldustarkvara Git hoidlast. Lahendus on testitav aadressil <http://nipt.herokuapp.com/> kasutajanimega „admin@admin“ ning parooliga „Fahrenheit451“. Näidisraport on leitav patsiendi isikukoodi „1111111111“ alt.

Eesti ID-kaardil põhineval isikutuvastusel on vajalik serveri seadistamine ning iga testija isikukoodi andmebaasi sisestamine. Lõputöö kirjutamise ajal ei ole Tartu Ülikooli virtuaalmasinate seadistamine lõpule jõudnud, mistõttu ei ole ID-kaarti toetav testrakendus saadaval. ID-kaardi põhine isikutuvastus on testitav lokaalselt. Instruktsioonid testimiseks on esitatud peatükis „Juhised“.

3. Nõuded

Käesolev peatükk esitab rakendusele püstitatud nõuded.

Ainus mittefunktsionaalne nõue on, et veebirakendus peab käivituma operatsioonisüsteemil Linux ning riistvaral, millel on 1 GB RAM ja 40 GB kettaruumi.

Järgnevad alapeatükid esitavad tarkvarale püstitatud funktsionaalsed nõuded.

3.1 Patsiendi ülevaatus

Käesolev alapeatükk keskendub patsiendi andmete sisestamisele süsteemi.

Arst võtab lapseootel naiselt vereproovi ja täidab seejärel alapeatükis käsitletud vormi.

Vormi disainimisel esitas Tervise-TAK väljad, mis on vajalikud geenitesti jaoks. Vormi disainimisel osalesid naistearst Dr Eva-Liina Ustav ja meditsiinigeneetik Dr Tiia Reimand.

Püstitatud funktsionaalsed nõuded:

- PÜ1. Ainult kasutajad, kellel on õigus patsiendi ülevaatus luua, pääsevad ligi patsiendi ülevaatusvormi loomise leheküljele.
- PÜ2. Ainult kasutajad, kellel on õigus patsiendi ülevaatus luua, saavad üles laadida patsiendi ülevaatusvormi.
- PÜ3. Vormis peavad olema järgmised väljad (tabel 1):
1. „Patsiendi isikukood“;
 2. „Vereproovi kood ja võtmise aeg“;
 3. „Etniline päritolu“;
 4. „Kehamassiindeks enne rasedust“;
 5. „Raseduse nädal vereproovi võtmisel“;
 6. „Loodete arv“;
 7. „Teadaolev pahaloomuline kasvaja“;
 8. „Luuüdi transplantatsioon“;
 9. „IVF rasedusel kasutatud doonormunarakku“;
 10. „Arsti e-post“.
- PÜ4. Vormi täites peavad väljade väärtused vastama tabelis 1 välja toodud oodatavale sisendile.
- PÜ5. Tabelis 1 välja toodud sisendi vastu eksides peab süsteem vormi üleslaadimist takistama ja teavitama kasutajat ebasobiva sisendiga väljast.

Tabel 1. Patsiendi andmete süsteemi sisestamisel täidetavad väljad ja nende oodatud sisendid.

Välja ID	Väli	Oodatud sisend
1	Patsiendi isikukood	Isikukoodi moodustamise algoritmile vastav isikukood [10]
2	Vereproovi kood ja võtmise aeg	Maksimaalselt 100 tähemärgist koosnev sisend, mis ei esine varasemalt andmebaasis
3	Etniline päritolu	Valik rippmenüüst
4	Kehamassiindeks enne rasedust	Positiivne ujukomaarv
5	Raseduse nädal vereproovi võtmisel	Positiivne täisarv
6	Loodete arv	Valik „Üks“
7	Teadaolev pahaloomuline kasvaja	Valik „Ei“
8	Luuüdi transplantatsioon	Valik „Ei“
9	IVF rasedusel kasutatud doonormunarakku	Valik „Ei“
10	Arsti e-post	Korrektne e-posti aadress, mis asub andmebaasis

Tabelis 1 väljade 4–9 erinevusel oodatud sisendist ei ole võimalik testi sooritada. Väli 3 annab lisainfot juhul, kui test on ebaõnnestunud. Väli 2 peab tagama, et vereproovi käitlemisel, mis läheb vastuollu proovi kogumise protokolliga, pööratakse tähelepanu ja edasi ei analüüsita.

Kasutuslugu patsiendi andmete süsteemi sisestamisest on esitatud lisa III.

3.2 Raporti loomine

Raporti loomise alapeatükis käsitletakse olukorda, kus testitava patsiendi vormiandmed ja geenitesti tulemused on andmebaasis saadaval.

Raporti loomiseks siseneb süsteemi eksperthindaja, kes andmebaasis oleva informatsiooni põhjal kinnitab loote normaalarengu või võimaliku väärarengu. Õige diagnoosi määramiseks peab süsteem kuvama kogu teadaoleva informatsiooni patsiendi kohta.

Pärast raporti sisestamist andmebaasi teavitab süsteem patsiendi kontaktisikuks määratud arsti tulemuste olemasolust.

Raporti loomisele püstitatud funktsionaalsed nõuded:

- RL1. Raportit saavad luua ainult kasutajad, kellel on õigus raporti loomiseks süsteemis.
- RL2. Raporti loomisel peab veebiliides kuvama patsiendi ülevaatusel vormi sisestatud andmed.
- RL3. Raporti loojale peab veebiliides kuvama peatükis „Andmemudel“ käsitletud relatsioonide *examination*, *db_data*, *run*, *chromosome* kõigi tunnuste väärtused.
- RL4. Raporti looja peab saama kirjutada hinnangu vabas tekstis tähemärkide arvuga 5000.
- RL5. Raporti looja peab saama valida otsuse „ebanormaalne“ ja „normaalne“ vahel.
- RL6. Pärast raporti kinnitamist peab tarkvara saatma teavituse e-posti aadressile, mis oli märgitud hinnatava patsiendi vormis välja „Arsti e-post“ all (tabel 1).

Kasutuslugu raporti loomisest on esitatud lisas IV.

3.3 Raporti vaatamine

Käesolev alapeatükk keskendub situatsioonile, kus patsiendi andmetel põhinev raport on eksperthindaja poolt loodud ja arsti on teavitatud tulemuste olemasolust e-posti teel.

Raportist näeb arst patsiendi geenitesti tulemusi, mille põhjal langetab arst edasise otsuse.

Raporti vaatamisele on esitatud järgnevad funktsionaalsed nõuded:

- RV1. Raportit saavad vaadata ainult raporti vaatamise õigusega kasutajad.
- RV2. Raportit saab alla laadida PDF-failina.
- RV3. Raportis on näha eksperthindaja otsus.
- RV4. Raportis on näha kõik need väljad, mis olid hindamise hetkel nähtavad.

Kasutuslugu raporti alla laadimisest on esitatud lisas V.

3.4 Kasutajate lisamine

Peatükis käsitletakse kolmandatele osapooltele süsteemi ligipääsemise võimaldamist.

Selline olukord võib tekkida, kui arst X on vahemikus Y töölt eemal, kuid patsiendi Z tulemused saabuvad vahemikus Y, ent teine arst, W, on sellel ajal saadaval. Sellisel juhul saab X luua W kasutaja, et W saaks Z tulemustele reageerida.

Kasutajate lisamisele püstitatud nõuded:

- KL1. Iga kasutaja, kellel on õigus kasutajaid lisada, saab luua uue kasutaja.
- KL2. Uut kasutajat luues tuleb sisestada kasutaja e-posti aadress, eesnimi, perenimi, isikukood ja määrata ära kasutaja õigused.

- KL3. Kasutaja looja saab valida uuele kasutajale alamhulga endal olemasolevatest õigustest.
- KL4. Kasutaja loomise õnnestumisel saadetakse uue kasutaja e-posti aadressile parool ja kasutajatunnus, millega saab uus kasutaja sisse logida.

Kasutuslugu kasutajate lisamisest on esitatud lisan VI.

3.5 Üldised nõuded

Järgnevas alapeatükis tuuakse välja nõuded, mis hõlmavad teatud määral kõiki varasemaid peatükke, ent on liiga spetsiifilised, et neid käsitleda varem.

Üldisemad funktsionaalsed nõuded:

- ÜN1. Veebileidese funktsionaalsusele saavad ligi ainult autenditud kasutajad.
- ÜN2. Kasutaja peab saama parooli muuta.
- ÜN3. Kasutaja peab saama muuta nende kasutajate õigusi, kellele ta on süsteemi ligipääsu andnud.
- ÜN4. Veebileidese keelt saab valida kolme keele vahel: eesti, inglise ja vene.

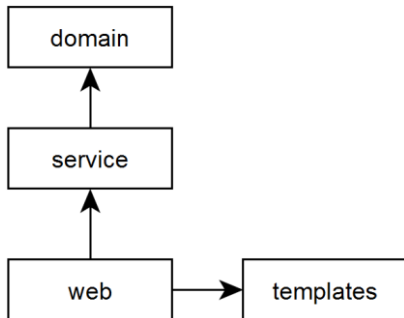
Nõudeid ÜN2 ja ÜN3 katvad kasutuslood on esitatud vastavalt lisades VII ja VIII.

4. Arhitektuur

Käesolevas peatükis kirjeldatakse valminud lahenduse arhitektuuri. Esimene alapeatükk keskendub veebirakendusele ja teine lahendusele tervikuna.

4.1 Veebirakenduse arhitektuur

Autor lähtub MVC (*Model-View-Controller*) [11] struktuurist, kuhu on eraldi lisatud teenuse kiht ja projektifailide paigutamisel Spring Boot projekti struktuuri soovitusel (joonis 2) [12].



Joonis 2. Valminud tarkvara arhitektuur, mille loomisel lähtuti *Model-View-Controller* struktuurist. Kontrollerid paiknevad paketi *web*, teenused asuvad paketi *service*. Mudel asub paketi *domain*. Vaated (*view*) on esindatud ressursside kataloogis *templates*.

MVC korral on kõik kihid eraldiseisvad ja seetõttu on tulevikus muudatuste sisseviimine süsteemi lihtsam, sest üksteisest sõltuvaid komponente on vähem [11]. Autor otsustas lisaks eelnevale kasutada andmeedastusobjekte (*data transfer object*) andmete edastamiseks teenustest kontrolleritesse. See vähendab kontrolleri ja mudeli sõltuvust üksteisest, sest kontroller ja mudel ei kasuta enam samu klasse. Teisendamise tagab teenuse kiht, mistõttu muudatuste korral ei ole vaja tõenäoliselt muuta kõiki kihte.

Autor proovis kasutada võimalikult palju abimeetodeid, mis oleksid eraldiseisvalt testitavad ning toetaksid lõimtöötlust. Selleks lõi autor paketi *util*, mis sisaldab klasse staatiliste abimeetoditega, mida kasutavad mitmed teenused veebirakenduses. Testide kirjutamine nendele staatilistele meetoditele on lihtne, sest ükski meetod ei eelda ühendamist andmebaasiga ega veebirakenduse käivitamist.

Rakenduse kirjutamist hõlbustas Lombok, mis asendas tihti kasutatavate isoleeritud Java koodijuppide kirjutamise annotatsioonide kirjutamisega [13].

Süsteemis on järgnevad teenused:

1. *ExaminationService* vastutab patsiendi andmete süsteemi sisestamise eest.
2. *MailService* vastutab e-posti saatmise eest.
3. *ResultsService* tagab raporti loomise ja allalaadimisega seotud funktsionaalsuse.
4. *UserService* vastutab kasutajatega seotud funktsionaalsuse eest.

Lisaks eelnevalt käsitletud pakettidele on olulisemad paketid veel:

1. Pakett *configuration* sisaldab rakenduse sätteid määravaid klasse. Turvalisusega seotud sätete määramist illustreerib klass *SecurityConfigProduction*.
2. Pakett *constraint* sisaldab autori loodud annotatsiooni, mis kontrollib isikukoodi korrektsust.
3. Paketis *db* asuvad liidesed, mis laiendavad klassi *JpaRepository* ning mida kasutatakse andmevahetuseks andmebaasiga.
4. Pakett *dto* hoiustab andmeedastusobjekte.
5. Pakett *feedback* sisaldab eesteenusesse teadete liigutamise loogikat.

Kaustas *resources* paiknevad HTML, CSS (*Cascading Style Sheets*), JavaScript, tõlke ning konfiguratsiooniga seotud failid.

4.2 Lahenduse arhitektuur tervikuna

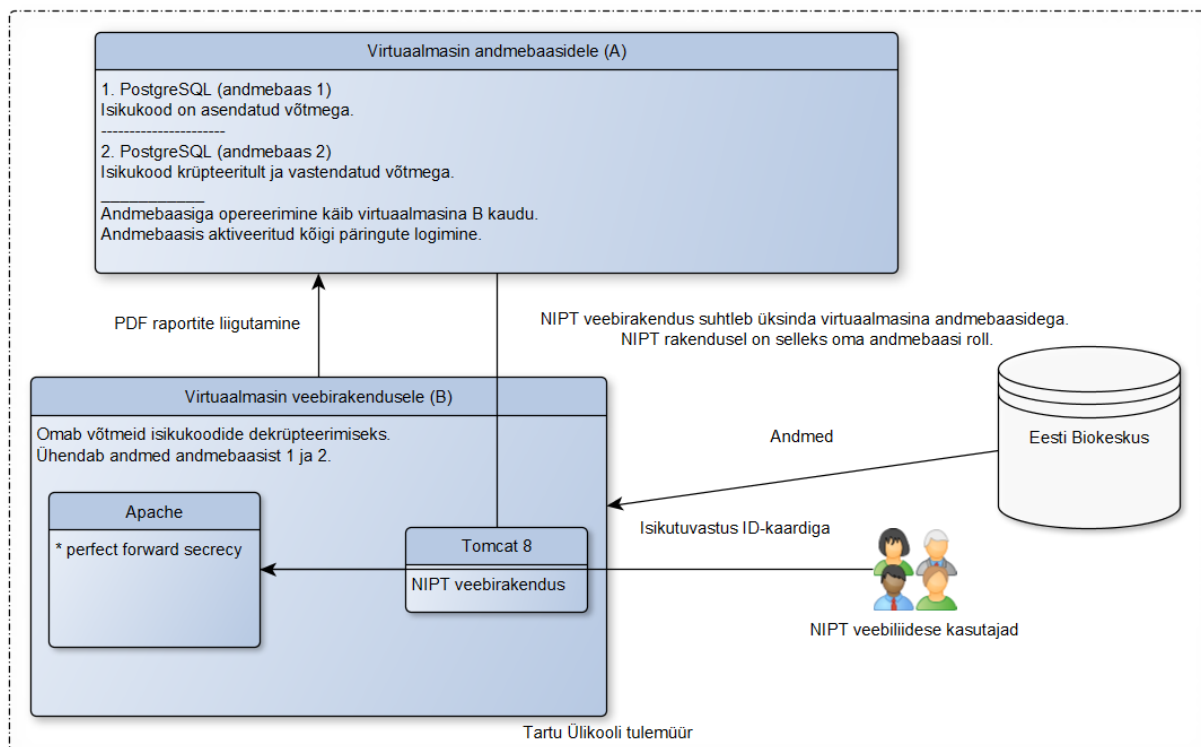
Veebirakendus ja andmebaasid hakkavad töötama Tartu Ülikooli tule müüri taga (joonis 3). Toetudes Tartu Ülikooli Infotehnoloogia osakonna ja Kaarel Kitsemetsa (Twilio Inc., tarkvarainsener) nõuannetele, disainis autor võimalikult turvalise lahenduse (joonis 3).

Lahendus vajab kahte virtuaalmasinat, millest esimeses (virtuaalmasin A) töötavad kaks andmebaasi ja teises (virtuaalmasin B) veebirakendus. Virtuaalmasin A on krüpteeritud. Mõlemale virtuaalmasinale pääseb ligi üks administraator turvakestega (*Secure Shell*). Meldimiseks kasutatakse asümmeetrilist krüptograafiat. Virtuaalmasinale A pääseb lisaks administraatorile ligi virtuaalmasin B.

Mõlemad andmebaasid virtuaalmasinas A logivad iga päringu. Iga andmebaasi kasutaja rollid on täpselt piiritletud. Virtuaalmasin B suudab ainsana dekrüpteerida andmebaasis olevaid krüpteeritud välju.

Veebirakenduse kasutajad pääsevad veebiliidesele ligi ainult ID-kaardiga. See vähendab kasutajakontode jagamist erinevate inimeste vahel.

Virtuaalmasin B vahendab andmeid Eesti Biokeskuse (asub Tartu Ülikooli tulemüüri taga) ja virtuaalmasina A vahel.



Joonis 3. Tarkvaralahendus Tartu Ülikooli tulemüüri taga. Veebirakendus paikneb virtuaalmasinas B. Andmebaasid töötavad virtuaalmasinas A. Virtuaalmasin B vahendab testi tulemusi virtuaalmasina A ja Eesti Biokeskuse vahel.

Lisaks andmebaasi, veebirakenduse ning operatsioonisüsteemi tasemel logimisele luuakse iga ülevaatus raporti loomisel raportist koopia ning edastatakse virtuaalmasinasse A.

Virtuaalmasinas B töötab Apache veebiserver, mis vahendab päringuid samas masinas Tomcat 8 peal töötavale veebirakendusele. Lisameetmena on aktiveeritud *perfect forward secrecy*, mis tagab, et serveri privaatse võtme lekkimisel ei ole varasem infovahetus dekrüpteeritav.

5. Insener-tehnilised valikud

Käesolev peatükk keskendub tarkvara tehnilistele valikutele, teekidele ning raamistikkudele, millega autor töö kirjutamise ajal kokku puutus.

5.1 Veebiraamistik

Veebiraamistiku valimisel oli oluline, et raamistik põhineks autori jaoks tuttavalt programmeerimiskeelele Java, leidsuks põhjalik dokumentatsioon, lai kasutajaskond ja veebirakenduse programmeerimine oleks mõistlikult teostatav.

Eelnevatele kriteeriumitele tuginedes jäid valikusse Spark Java ja Spring Boot.

Autor alustas veebirakenduse kirjutamist Java 8 põhineval Spark Javal, sest oli sama raamistikku varem kasutanud [14]. Spark Java on lihtsusele suunatud mikroraamistik [14], mida illustreerib nende kodulehel asuv näiterakendus, mis koosneb vähem kui kuuest koodireast [14], ning kasutajasõbralik dokumentatsioon [15]. Lisaks on Spark Java veebilehel mitmeid õpetusi erinevate ülesannete lahendamiseks [16].

Üsna varsti selgusid Spark Java kui mikroraamistiku puudujäägid. Esmapilgul kasutajasõbralik dokumentatsioon osutus puudulikuks ja pealiskaudseks. Malli mootorite ja kasutajate autentimise dokumentatsioon puudus ning viidati vaid GitHubi koodinäidetele [15].

Autori hinnangul ei olnud raamistik turvaline, sest õpetustes leiduv näide autentimist kasutavast rakendusest põhines autentimissüsteemi käsitsi ehitamisel [17], mis avab tee võimalikele turvaaukudele.

Kõigele eelnevale lisaks puudusid raamistikul Java Spark andmebaasiga suhtlemise abstraktsioon, ühtne konfiguratsioonifail ja e-posti liides. Oli selge, et arendatav veebirakendus oli liiga keeruline mikroraamistiku jaoks ning autor hakkas kaaluma raamistiku vahetamist.

Spring Boot on Javal põhinev veebiraamistik, millel on oluliselt põhjalikum dokumentatsioon [18] võrreldes Java Spark dokumentatsiooniga. Eelisteks on lai kasutajaskond, hulganisti näiterakendusi [19], integreeritud autentimissüsteem [20], liides e-posti [21] ja andmebaasiga suhtlemiseks [22], keskne konfiguratsiooni võimalus [23] ja integreeritud Thymeleaf malli mootori lahendus [24]. Osutus, et Spring Boot lahendab esinenud Java Sparki puudujäägid, mistõttu autor kirjutas tarkvara Spring Boot raamistikule.

5.2 Porditava dokumendiformaadi genereerija

Pärast tulemuste kinnitamist peab arstini jõudma raport PDF-kujul. PDF-genereerimisteegi valimisel oli esmatähtis vabavaralisus, abimaterjalide olemasolu ning võimekus luua XHTML-failist (*Extensible HyperText Markup Language*) PDF. Neid tingimusi silmas pidades jäid alles Flying Saucer ja Apache PDFBox.

Mõlemad teegid toetavad PDF genereerimist [25] [26]. Paraku Apache PDFBox kodulehelt ei õnnestunud leida arusaadavaid juhendeid ja kinnitust HTML-failist PDF genereerimise toest. Lõputöö kirjutamise hetkel oli nende näidete leht valmimisel, kus viidati koodihoidlale, mille näited ei olnud autori jaoks piisavalt arusaadavad [27]. Seetõttu langes Apache PDFBox võimaliku valikuna välja. Flying Saucer seevastu on hästi dokumenteeritud teek, mis võimaldab XHTML põhjal genereerida PDF, millele on rakendatud CSS [28]. See lahendus sobis süsteemiga hästi, sest raporti leheküljel XHTML kujul oli kirjutamise hetkel juba olemas, seega piisas ainult teegi lisamisest. Nende GitHubi leheküljel leidis piisavalt näiteid, kuidas teeki kasutada [26]. Seetõttu valis autor Flying Saucer teegi.

5.3 ID-kaart

Autor ei olnud enne töö kirjutamist Eesti ID-kaardi põhise isikutuvastusega kokku puutunud. Sertifitseerimiskeskuse veebilehel asuvate õpetuste hulgas leidis juhendeid IIS, Apache, NGINX serveri ning PHP-põhise lahenduse kohta, ent mitte programmeerimiskeelele Java põhineva veebirakenduse jaoks [29].

Sertifitseerimiskeskuse väheste juhendite tõttu proovis autor Ideelabori autentimisprotokollil OpenID põhinevat teenust¹, mis võimaldab ID-kaardi põhise autentimist [30]. Paraku selline lähenemine sõltub Ideelaborist. Ideelabori teenus ei olnud alati saadaval, mistõttu autor loobus OpenID kasutamisest.

Sertifitseerimiskeskusest vastati autori päringule, et ID-kaardi põhise isikutuvastuse näol on tegu levinud kliendi sertifikaadil põhineva autentimisega. Täpsemal uurimisel osutus, et ID-kaardile on salvestatud kaks X.509 standardil põhinevat sertifikaati [31].

Autor leidis Spring Boot X.509 põhineval kliendi sertifikaadi isikutuvastuse näidislahenduse. Näidislahenduse põhjal piisas autoril Springi konfiguratsiooni muutmisest ja failide *keystore.jks* (sisaldab serveri privaateid võtmeid) ning *truststore.jks* (sisaldab avalikke võtmeid, mida usaldada) lisamisest, et aktiveerida sertifikaadipõhine isikutuvastus [32]. ID-

¹ <https://openid.ee/>

kaardi toe lisamiseks juhendus autor Sertifitseerimiskeskuse veebilehel asuva Apache konfigureerimise õpetusest ning lõi faili *truststore.jks* juhendis esitatud sertifikaatide põhjal. Protsessi käigus selgus, et igale kliendisertifikaadile pidi vahetult järgnema sertifikaadi väljaandja. Eelneva kahe juhendi põhjal õnnestus autoril realiseerida ID-kaardi põhine isikutuvastus.

5.4 E-post

E-posti saatmiseks kasutas autor Spring Boot e-posti liidest [21] ja realiseeris liidese teegiga JavaMail [33]. Teisi teeke autor ei kaalunud, sest Oracle'i pakutav JavaMail on pidevalt uuendatav ning Oracle'i puhul eeldab autor, et teek on testitud ja seetõttu ka töökindel.

5.5 Kujundus

Autor kaalus kolme lahendust kujunduse pakkujate osas. Esimene variant oli autori enda CSS loomine. Teine variant oli ZURB Foundation ja kolmas alternatiiv Bootstrap.

Pidades silmas asjaolu, et veebirakendus peab ka ilus välja nägema ning võiks toetada erinevaid ekraanisuurusi ja veebilehitsejaid, sai autorile selgeks, et kujunduse manuaalne kirjutamine on aeganõudev ja ei tule esteetiline välja, sest autoril puudub kogemus CSS kirjutamisega.

Järele jäid variantidena ZURB Foundation ja Bootstrap. Mõlemad raamistikud on väga levinud ja pakuvad sarnaseid võimalusi. Nii Foundation kui ka Bootstrap mõlemad toetavad kõiki levinumaid veebilehitsejaid ja teatud määral ka Internet Explorer 8-t [34] [35].

Autor proovis mõlemaid teeke, kuid paraku tekkis probleeme Foundationi navigeerimisribaga, mida oli väga keeruline korrektselt kuvama saada. Bootstrapiga ei esinenud ühtegi probleemi. Kõigile tekkinud küsimustele leidis autor kiiresti lahenduse Bootstrapi kodulehelt või mõnelt muult programmeerimisalaselt veebilehelt.

Bootstrapi hoidlat jälgib sada tuhat GitHubi kasutajat [36], mida on üle nelja korra rohkem kui Foundationi hoidlal [37]. Seega on raamistikuga Bootstrap tekkivatele probleemidele lahenduse leidmise tõenäosus suurem kui Foundationi puhul. Eelneva põhjal valis autor raamistiku Bootstrap.

5.6 Vormi valideerimine

Patsiendi andmete valideerimise süsteemi sisestamisel teostas autor Spring Boot juhendi järgi [38]. Juhendi põhjal määratakse andmeedastusobjekti, mis tagastatakse andmete postitamisel tagateenusesse (*back end*), isendiväljadele piirangud annotatsioonidega [38]. Sellise

lähenedes korral ei ole vaja kasutaja sisendit manuaalselt kontrollida ning programmikood on seetõttu lihtsamini loetav.

Lisaks olemasolevatele piirangutele kirjutas autor piirangu Eestis välja antud isikukoodi valideerimiseks. Ei saa eeldada, et testitavale on väljastatud Eesti Vabariigi isikukood. Seetõttu lõi autor süsteemi võimaluse aktsepteerida lisaks Eestis väljastatud isikukoodile identifikaatorit, mis koosneb kolmest järjestikusest komponendist: suvalisest numbrite järjestist, märgisest „A“ ja sünnikuupäevast kujul ppkkaaaa. Näiteks 2. septembril 1995 sündinud inimesele võib vastata märgis „123456A02091995“.

5.7 Kasutajaõigused

Tegu on veebirakendusega, mis tegeleb patsiendi isikuandmete töötlemisega. Autor pööras tähelepanu rakenduse turvalisusele, vältides turvalahenduste ise realiseerimist ja kasutades olemasolevaid Spring Boot testitud lahendusi [39].

Süsteemi kasutajal saavad olla õigused, mis on välja toodud tabelis 2.

Tabel 2. Õiguste süsteem valminud tarkvaras. Tarkvara kasutajal saab olla neli õigust, mis piiritlevad ligipääsu tarkvara funktsionaalsusele.

Õigus	Kirjeldus	Märkus
Raporti vaatamine	Võimaldab raportit kuvada ja alla laadida PDF-failina.	
Raporti loomine	Võimaldab raportit luua.	Kasutaja näeb informatsiooni, mis raportisse tuleb. PDF-faili alla laadida ei saa.
Patsiendi ülevaatusandmete sisestamine	Lubab minna ülevaatuslehele, andmed sisestada ja salvestada andmebaasi.	
Kasutajate lisamine	Lubab minna kasutajate leheküljele ja luua uusi kasutajaid.	Uuel kasutajal saavad olla maksimaalselt samad õigused, mis kasutaja loojal.

Pidades silmas funktsionaalseid nõudeid, mis defineerivad erinevatele kasutajatele erinevaid õigusi ning MVC struktuuri, pidas autor põhjendatuks piirata kasutajate juurdepääsu rakenduse funktsionaalsusele veebirakenduses meetodite tasemel [40].

Selleks kasutas autor annotatsiooni *PreAuthorize*, mille argumendiks on meetod, mis võrdleb kasutaja tegelikku õigust nõutud õigusega [41]. Kui õigus on olemas, siis jätkatakse annoteeritud meetodi täitmisega, vastasel korral kuvatakse kasutajale vealeht. Süsteemis kasutab autor *PreAuthorize* annotatsiooni kontrolleri meetoditel. Lahenduse puuduseks on tõik, et kui mõnele meetodile jääb annotatsioon lisamata, siis pääsevad funktsionaalsusele ligi kõik sisse loginud kasutajad.

5.8 Tõlge

Autor kasutas mallimootori Thymeleaf võimalust teksti fragmente mallist välja viia, muutes HTML-lehtedele tõlke lisamise lihtsaks [42].

HTML-failis tuleb defineerida väli *Message Expressions*, kuhu kirjutatakse võtmesõna. Võtmesõnale peab vastama eeldefineeritud tõlkefailis tekst. Käitusajal asendatakse HTML-failis *Message Expressions* väli tõlkefailis asuva tekstiga. Spring Boot konfiguratsioonist sõltub, millisest tõlkefailist vaste valitakse.

Konfiguratsioonis oli oluline määrata kodeeringuks UTF-8 ning seejärel tagada eesti- ja venekeelses tõlkefailis UTF-8 kodeering, sest vastasel juhul ei kuvatud kõiki tekstisümboleid korrektselt.

Autor programmeeris süsteemi, kus pärast kasutaja sisselogimist süsteemi laaditakse andmebaasist keeleinfo ja salvestatakse küpsisesse, pärast mida oskab Spring Boot kasutajale sobivas keeles veebiliidest esitada. Kasutajal on võimalik keelt veebiliideses vahetada.

Alternatiivne lahendus oleks olnud iga keele kohta eraldi HTML-faili kirjutamine, ent sellisel juhul tuleks väikese muudatuse korral igas HTML-failis muudatused teha ja seetõttu otsustas autor Thymeleaf tõlkelahenduse kasuks.

5.9 Andmebaas

Andmebaasi valides oli oluline, et andmebaas ei seaks hiljem piiranguid ja oleks vabavaraline.

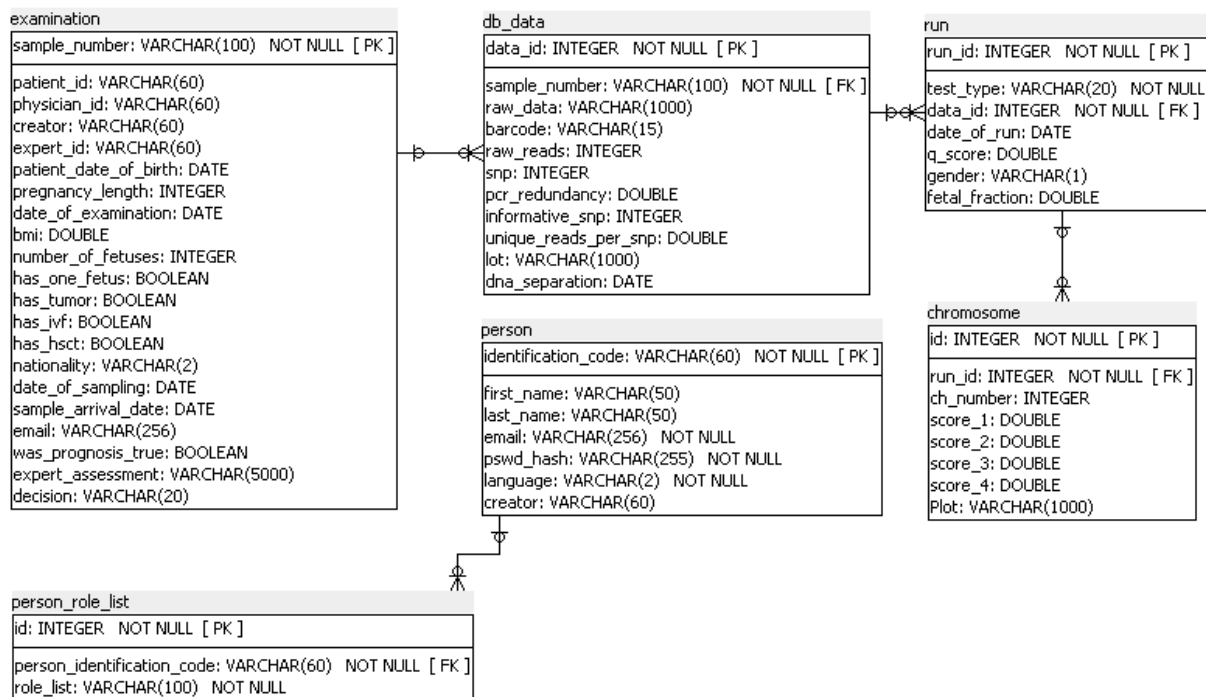
Esimene ja ühtlasi lõplik valik oli PostgreSQL 9,6. PostgreSQL andmebaasi suurus ei ole piiratud, ühe relatsiooni maksimaalne suurus on 32 TB, maksimaalne rea suurus on 1,6 TB, välja maksimaalne suurus on 1 GB, ridade arv relatsioonis ei ole piiratud ning maksimaalne arv lubatud veerge on 250-1600 sõltuvalt veergude tüüpidest [43]. Seega ei sea andmebaas sisuliselt piiranguid. Eelnevale lisaks on PostgreSQLil mahukas (3508 lk) dokumentatsioon [44].

PostgreSQL töötab kõigil levinumatel operatsioonisüsteemidel, sealhulgas ka Mac OS X peal [43].

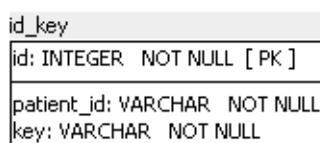
Andmete pärimiseks otsustas autor Spring Boot JpaRepository kasuks. Tegu on abstraktsiooniga, kus programmeerija ei pea ise realiseerima andmebaasi päringuid [22]. Alternatiivne lahendus oluks manuaalselt päringute kirjutamine, ent aja säästmiseks ja lihtsama koodi nimel valis autor olemasoleva lahenduse.

5.10 Andmemudel

Tulevikus hakkab tarkvara turvalisuse kaalutlusel paralleelselt pöörduma kahe andmebaasi poole. Ühes andmebaasis on joonisel 4 esitatud andmemudel ja teises andmebaasis joonisel 5 esitatud mudel. Testversioon kasutab ainult ühte andmebaasi ja joonisel 4 esitatud mudelit, kus välja *patient_id* all asub patsiendi isikukood muutmata kujul.



Joonis 4. Andmemudel, mis hoiustab endas patsiendi testiga seotud informatsiooni ja süsteemi kasutajate andmeid.



Joonis 5. Andmemudel, mis asub erinevas andmebaasis joonisel 4 esitatud mudelist. Mudel hoiab krüpteeritud patsiendi isikukoodi ja võtit, mis seob mudeli joonisel 4 esitatuga.

Examination (joonis 4) väli *patient_id* hakkab tulevikus hoiustama võtit, mis seob relatsiooni *examination* relatsiooni *id_key* väljaga *key*. Patsiendi isikukood hakkab krüpteeritult paiknema relatsiooni *id_key* välja *patient_id* all. Kahe andmebaasi kasutamine hajutab süsteemi ründe korral riski isikuandmete isikustatud kujul lekkimist. Hetkel ei ole kahe andmebaasi kasutamine realiseeritud, sest kirjutamise hetkel ei ole Tartu Ülikooli virtuaalmasinate seadistamine lõpuni jõudnud. Seetõttu ei ole veel teada, kuidas krüpteerimiseks ning dekrüpteerimiseks vajalike võtmete haldus töötama hakkab.

Geenitesti seisukohalt on kõige olulisemad relatsioonid (joonis 4) *examination*, *db_data*, *run* ja *chromosome*.

Relatsioon *examination* sisaldab endas informatsiooni, mis küsitakse patsiendilt ülevaatusel, ja informatsiooni lõpptulemuste kohta. Ühtlasi hoiustatakse seal ülevaatusel andmed süsteemi postitanu ja eksperthindaja isikukoodi.

Relatsioon *examination* on üks-mitmele seoses relatsiooniga *db_data*, mille atribuudid on seotud patsiendi geenitesti tulemustega.

Relatsiooniga *db_data* on üks-mitmele seoses relatsiooniga *run*, mis hoiab endas üldisemat informatsiooni geenitesti kohta. Relatsioonis on atribuudid, millal ja missugust bioinformaatilist analüüsi kasutati.

Relatsioon *run* on üks-mitmele seoses relatsiooniga *chromosome*, mis sisaldab hinnanguid patsiendi analüüsitud kromosoomide kohta. Relatsiooni *chromosome* sisestatakse iga kromosoomi kohta hinnang, mille geenitest väljastab. Tihti on väljastatavaks hinnanguks arvuline väärtus, mille interpretatsioon sõltub andmeanalüüsi meetodist.

Andmebaasi üks-mitmele relatsioonide seosed tagavad, et kui mõnes protsessi etapis tekib vajadus eelnevat etappi korrata, siis saab seda nii teha, et mõlema etapi andmed jäävad alles.

Veebirakenduse arendamisel osutus, et ei piisa ainult autenditud kasutajate (joonis 4, relatsioon *person*) ja sisse logimata kasutajate eraldamisest. Vältimaks kasutajal, kes kasutab süsteemi ainult patsiendi andmete sisestamiseks, näha patsiendi diagnoosi, oli vaja luua õiguste süsteem. Kasutaja õigusi esindab relatsioon *person_role_list*. Igal kasutajal saab olla mitu õigust.

Andmemudelist leidub üle kuue erineva versiooni, sest erinevatel arendusetappidel esines erinevaid väljakutseid. Ühel etapil disainis autor andmemudeli anonüümsuse põhimõttel, kus andmebaasi olemit ei ole võimalik testi pakkujal patsiendiga reaalmaailmas seostada. Igal tellitaval testil oli unikaalne identifikaator, mis oli teada ainult arstile ning sidus testi ja

patsiendi. Disaini stiimuliks oli lootus, et lähenemine vähendab isikuandmete töötlemisest tulenevaid kohustusi, kuid see ei realiseerunud. Lisandus hoopis üleliigne keerukus ning autor loobus sellisest lähenemisest.

Andmemudeli esimestel versioonidel oli mitmeid puudujääke. Näiteks relatsiooni *chromosome* asemel olid relatsioonid spetsiifilise kromosoomi jaoks: *chromosome2*, *chromosome3*, *chromosome13*, *chromosome18* ja *chromosome21*. Kõige viimases versioonis on üksainus relatsioon *chromosome*, milles on atribuudiks kromosoomi number, mis võimaldab sisestada andmeid rohkemate kromosoomide kohta kui 2, 3, 13, 18 ja 21.

Inimeksimuste vähendamist silmas pidades proovis autor andmebaasi relatsiooni atribuutide nimedes ja andmetüüpides olla semantiliselt võimalikult täpne. Näiteks väli *decision* (joonis 4, relatsioon *examination*) oli andmebaasi arendamise alguses arvu tüüpi, kus „0“ tähistas varianti „ebanormaalne“ ja „1“ varianti „normaalne“. Paraku on see eksitav, sest „0“ saab mõelda kui „trisoomiat ei ole“, mistõttu kasutab autor võimalikult palju andmetüüpe *variable character field* ja loogikamuutujat (*boolean*), mis on üheselt mõistetavad.

6. Juhised

Järgnevas peatükis tuuakse välja rakenduse käivitamise juhised ja minimaalsed nõuded riistvarale ja tarkvarale, et rakendust testida.

Lokaalseks testimiseks on vaja Linuxil põhinevat arvutit, millel on 512 MB RAM, 300 MB vaba kettaruumi veebirakendusele kasutamiseks ja internetiühendus. Tabelis 3 on välja toodud vajalikud juhised rakenduse käivitamiseks. Pärast tabelis 3 esitatud täitmist pääseb veebirakendusele ligi aadressil <http://localhost:8080/> kasutajanimel „admin@admin“ ning parooliga „Fahrenheit451“. Näidisraport on leitav patsiendi isikukoodi „1111111111“ alt.

Tabel 3 - Juhised veebirakenduse käivitamiseks operatsioonisüsteemil Ubuntu 16.04.

Tegevuse järjekord	Kirjeldus	Käsk
1	Installeerib JDK 8.	<code>sudo apt install openjdk-8-jdk</code>
2	Installeerib versioonihaldustarkvara Git.	<code>sudo apt install git</code>
3	Kloonib projekti	<code>git clone https://github.com/PriitPaluoja/thesis.git</code>
4	Navigeerib projekti kausta.	<code>cd thesis</code>
5	Lisab käivitamisõiguse.	<code>chmod +x deploy-local.sh</code>
6	Lisab käivitamisõiguse.	<code>chmod +x gradlew</code>
7	Käivitab rakenduse.	<code>./deploy-local.sh</code>

Tabelis 3 esitatud juhiste täitmise järel pöördub veebirakendus teenuses Heroku paikneva testandmebaasi poole.

Eesti ID-kaardi testimiseks on vajalik ID-tarkvara². Faili `application.properties` sisu tuleb asendada faili `application.properties.id` sisuga. Pärast tabelis 3 esitatud juhiste täitmist aadressil <https://localhost:8443/> nõuab veebiliides kasutajalt ID-kaardi põhist autentimist.

² <https://installer.id.ee/>

7. Kvaliteet ja nõutele vastavus

Käesolev peatükk annab ülevaate, kuidas tagati kvaliteet ja veebirakendusele esitatud nõuded.

Autor kirjutas programmeerimisel võimalikult palju abimeetodeid, mis paiknevad paketi *util*. Paketi *util* olevad klassid ja nende staatilised meetodid on lihtsasti testitavad. 19 testimise meetodit tagavad paketi *util* kõigi klasside meetodite korrektse töötamise.

Pärast projekti valmimist tegi autor projektile staatilist koodi analüüsi pistikprogrammiga (*plugin*) FindBugs, mis tuvastas 48 *malicious code vulnerability* kategooriaga programmiviga. Vead olid seoses andmeedastusobjekti ja andmebaasiga seotud meetoditega, mis muutsid või tagastasid muteeritavaid isendivälju. FindBugs programmivigade kirjelduses soovitati need parandada juhul, kui neid klassi instantsi kasutab väline tarkvara [45]. Kuna sellist olukorda ei teki, siis autor ignoreeris hoiatusi.

Autor kommenteeris kõik liidestest deklareeritud meetodid ja kontrollis manuaalselt üle teenusklasside loogika.

Kõik kasutajale veebis kuvatavad HTML-failid läbisid W3C (World Wide Web Consortium) HTML validaatori kontrolli ilma vigadeta.

Töös valminud veebirakendusele püstitati funktsionaalseid nõudeid viiest erinevast kategooriast: patsiendi ülevaatus (PÜ), raporti loomine (RL), raporti vaatamine (RV), kasutajate lisamine (KL) ja üldised nõuded (ÜN).

Nõuded PÜ1, PÜ2, RL1, RV1 ja ÜN1 tagas autor Spring Boot sisseehitatud autentimis-süsteemiga, kus autor aktiveeris konfiguratsiooni, milles enne *PreAuthorize* annoteeritud kontrolleri meetodite poole pöördumist kontrollib süsteem vastavalt meetodile, kas kasutajal on olemas sobiv õigus kontrolleri meetodi poole pöördumiseks. Autor verifitseeris manuaalselt, kas süsteem keelab ilma õiguseta nendele meetodile ligipääsu või mitte. Süsteem vastas nõuetele.

PÜ3–PÜ5 vastavuse tagas autor Spring Boot vormi valideerimise õpetuse järgi, kus andmeedastusklassi isendiväljadele määrati piirangud ja Thymeleaf süntaksiga defineeriti ära käitumine vigase sisendi korral. Autor veendus piirangute töötamises manuaalselt.

Nõuded RL2–RL6, RV2–RV4, KL1–KL4 ja ÜN1–ÜN4 vastavust verifitseeris autor käsitsi.

Autor veendus lisades esitatud kasutuslugude vastavuses valminud tarkvarale.

Mittefunktsionaalsed nõuet testis autor Ubuntu 16.04.1 LTS virtuaalmasinas, kasutades Oracle VM VirtualBoxi virtualiseerimiseks. Autor määras virtuaalmasinale 1 GB RAM ja 40 GB kettaruumi. Hostmasinal oli 12 GB RAM, Core i5-4200U ning SSD.

8. Edasiarendusvõimalused

Järgnev peatükk annab ülevaate täiendavatest ülesannetest, mis on vaja teostada rakenduse praktikas kasutamiseks.

Kõige olulisem ülesanne on logimise lisamine. Autor plaanib kõik kasutajate tegevused logida turvalisuse kaalutlusel ning säilitada andmebaasis iga raporti varasemad versioonid. Raportite erinevate versioonide säilitamine tagab tulevikus diagnoosimise või tulemuste probleemide korral võimaluse ajaloo järgi uurida, mis ja kus valesti läks.

Ühtlasi on vajalik täiendav testimine ja rakenduse ülesseadmine Tartu Ülikooli tulemüüri taha.

Autoril on plaanis lisada avalehele ülevaade eksperthindajale tema hinnatud ja hindamisele kuuluvatest patsientidest. Analoogne lehekülg, mis annab ülevaate hinnatud ning hindamisel olevatest patsientidest, on vajalik ka arstile. Selline täiendus veebileidesele kiirendab kasutajate tööd, sest hetkel peavad kasutajad raportitega töötamiseks eelnevalt teadma patsientide isikukode.

Hetkeseisuga laaditakse joonised raportisse lingina väliselt aadressilt. Autoril on plaan muuta jooniste laadimine veebilehelt andmebaasist laadimiseks.

Eelnevad edasiarendusvõimalused on vajalikud praktiliseks kasutamiseks. Vähemoluliste täienduste hulka kuulub PDF-raportite esteetilisemaks muutmine, e-postil põhinevate teavituste sõnastuse redigeerimine ning uutele kasutajatele parooli edastamisviisi muutmine.

Nimetatud täienduste realiseerimine jäi lõputöö mahust välja, mistõttu valmis testversioon.

9. Kokkuvõte

Bakalaureusetöö eesmärgiks oli luua tarkvara (i) NIPT geenitesti elektrooniliseks tellimiseks, (ii) raporti genereerimiseks ja (iii) tulemustest automaatselt teavitamiseks.

Valminud lahenduse keerukus peitub järgnevas:

1. Lahendus töötleb **sensitiivseid terviseandmeid**, mis eeldavad turvalist disaini.
2. Loodud tarkvara on **multidistsiplinaarne**, ühendades geenitesti teostamiseks vajalikud etapid.
3. Tarkvara võimaldab geenitesti tulemusi **PDF-kujul** alla laadida.
4. Tarkvara kasutab Eesti **ID-kaardil** põhinevat isikutuvastust.
5. Veebiliidese **kasutajate õigused** on täpselt piiritletud.
6. Lahenduse veebiliides on **kolmekeelne**.

Tarkvara oli kirjutamise ajal uudne NIPT geenitesti tellimiseks, sest seni oli tellitud NIPT teste Eestis posti teel ja NIPT geenitesti pakkujate tellimissüsteeme ei olnud autorile teadaolevalt kirjeldatud. Lisaks nõuab iga uus analüüsimeetod oma andmebaasi struktuuri ja tarkvara funktsionaalsust.

Loodud tarkvara on oluline eelkõige seetõttu, et kiirendab tellimise protsessi ja vähendab vigade tekkimist andmete sisestamisel. Laiem eesmärk oli ressursi tõhusam kasutamine ning tervishoiuteenuse kvaliteedi tõstmine.

Valminud tarkvara põhineb veebiraamistikul **Spring Boot**, mallimootoril **Thymeleaf**, tihti kasutatavate isoleeritud Java koodijuppide genereerijal **Lombok** ja porditava dokumendi-formaadi genereerijal **Flying Saucer**. Veebiliidese kujundamiseks kasutati raamistikku **Bootstrap**. Andmete hoiustamiseks valiti **PostgreSQL** andmebaas. Töös esitati ning selgitati tarkvara andmemudeli disainipõhimõtted.

Arhitektuuris lähtuti *Model-View-Controller* struktuurist, et vähendada süsteemis üksteisest sõltuvuses olevate komponentide arvu. Lisaks võeti eeskujuks Spring Boot projekti struktuuri soovitusel. Tarkvaras kasutati võimalikult palju testidega kaetud abimeetodeid kvaliteedi tagamiseks.

Valminud veebiliides toetab **eesti**, **inglise** ja **vene** keelt. Kasutajate juurdepääs funktsionaalsusele on reguleeritud meetodite tasemel ning isikutuvastus on lahendatud **Eesti ID-kaardi** põhise autentimisega.

Töös esitati ka valminud lahenduse **arhitektuur** nii veebirakenduse tarkvara tasemel kui ka **virtuaalmasinate** tasemel.

Käesolevas töös esitati nõuded, millele peab valminud tarkvara vastama, ning veenduti nende vastavuses. Ühtlasi anti juhised rakenduse testimiseks ja toodi välja edasiarendusvõimalused.

Töö raames valmis testversioon. Lahendus vajab täiendavat testimist ja Tartu Ülikooli tulemüüri taha paigaldamist.

Valminud veebirakendus teeb võimalikuks geenitesti tellimise, raporti loomise ja tulemustest teavitamise tervishoiuteenuse pakkujale veebipõhiselt. Valminud testversioon läheb testimisse praktikas ning autor jätkab tarkvara arendamist väljaspool lõputööd.

Autor tänab

Suurimad tänusõnad juhendajatele Kaarel Krjutškovile, Priit Paltale ja Priit Adlerile. Suur tänu Olga Žilinale, kes aitas rakenduse vene keelde tõlkida, ning Kaarel Kitsemetsale, kes andis nõu turvalise disaini osas. Suur tänu Kaspar Paplile, Mari Liis Velnerile ning Hindrek Tederile soovitude ja toetuse eest ning dr Tiia Reimandile geenitesti tellimise näidisvormi jagamise eest.

Viidatud kirjandus

- [1] S. I. Nagaoka, T. J. Hassold ja P. A. Hunt, „Human aneuploidy: mechanisms and new insights into an age-old problem,“ 2012.
- [2] Sihtasutus Tartu Ülikooli Kliinikum Ühendlabor geneetikakeskus, „Sünnieelne diagnostika,“ [Võrgumaterjal]. <http://www.kliinikum.ee/geneetikakeskus/suennieelne-diagnostika>. [Kasutatud 29. märts 2017].
- [3] Sihtasutus Tartu Ülikooli Kliinikum Ühendlabor geneetikakeskus, „I trimestri sõeluuringud,“ Sihtasutus Tartu Ülikooli Kliinikum Ühendlabor geneetikakeskus, [Võrgumaterjal]. <http://www.kliinikum.ee/geneetikakeskus/suennieelne-diagnostika/13-itrimestri-soeluuringud>. [Kasutatud 29. märts 2017].
- [4] Sihtasutus Tartu Ülikooli Kliinikum Ühendlabor geneetikakeskus, „II trimestri sõeluuring,“ Sihtasutus Tartu Ülikooli Kliinikum Ühendlabor geneetikakeskus, [Võrgumaterjal]. <http://www.kliinikum.ee/geneetikakeskus/suennieelne-diagnostika/14-iitrimestrisoeluuring>. [Kasutatud 29. märts 2017].
- [5] Sihtasutus Tartu Ülikooli Kliinikum Ühendlabor geneetikakeskus, „Amniotsentees ja koorionibiopsia,“ Sihtasutus Tartu Ülikooli Kliinikum Ühendlabor geneetikakeskus, [Võrgumaterjal]. <http://www.kliinikum.ee/geneetikakeskus/suennieelne-diagnostika/17-amniotsentees-ac-kb-korionibiopsia>. [Kasutatud 29. märts 2017].
- [6] B. Zimmermann, M. Hill, G. Gemelos, Z. Demko, M. Banjevic, J. Baner, A. Ryan, S. Sigurjonsson, N. Chopra, M. Dodd, B. Levy ja M. Rabinowitz, „Non-invasive prenatal aneuploidy testing at chromosomes 13, 18, 21, X, and Y, using targeted sequencing of polymorphic loci,“ 2012.
- [7] D. Chitayat, S. Langlois ja R. D. Wilson, „Prenatal Screening for Fetal Aneuploidy in Singleton Pregnancies,“ kd. 33, nr 7, 2011.
- [8] H. Teder, *Evaluation of the possibility to detect fetal chromosome trisomies based on a defined set of single nucleotide polymorphisms for non-invasive prenatal testing*, Tartu, 2016.

- [9] Sihtasutus Tartu Ülikooli Kliinikum Ühendlabor geneetikakeskus, „NIPT,“ Sihtasutus Tartu Ülikooli Kliinikum Ühendlabor geneetikakeskus, [Võrgumaterjal]. <http://www.kliinikum.ee/geneetikakeskus/suennieelne-diagnostika/97-nipt>. [Kasutatud 29. märts 2017].
- [10] „Isikukoodide moodustamise, väljajagamise ja andmise kord - Riigi Teataja,“ 13 juuni 2015. [Võrgumaterjal]. <https://www.riigiteataja.ee/akt/838530?leiaKehtiv>. [Kasutatud 10. detsember 2016].
- [11] Tutorialspoint, „MVC Framework - Introduction,“ [Võrgumaterjal]. https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm. [Kasutatud 9. detsember 2016].
- [12] P. Webb, D. Syer, J. Long, S. Nicoll, R. Winch, A. Wilkinson, M. Overdijk, C. Dupuis ja S. Deleuze, „14. Structuring your code,“ [Võrgumaterjal]. <http://docs.spring.io/spring-boot/docs/current/reference/html/using-boot-structuring-your-code.html>. [Kasutatud 9. detsember 2016].
- [13] M. Kimberlin, „Software Engineering Tech Trends January 20,“ Object Computing, 20 jaanuar 2010. [Võrgumaterjal]. <http://jnb.ociweb.com/jnb/jnbJan2010.html#intro>. [Kasutatud 2. veebruar 2017].
- [14] Spark, „Spark Framework - A tiny Java web framework,“ [Võrgumaterjal]. <http://sparkjava.com/>. [Kasutatud 6. detsember 2016].
- [15] Spark, „Spark Framework - Documentation,“ [Võrgumaterjal]. <http://sparkjava.com/documentation.html>. [Kasutatud 6. detsember 2016].
- [16] Spark, „Spark Framework tutorials - Spark Java tutorials,“ [Võrgumaterjal]. <https://sparktutorials.github.io/>. [Kasutatud 6. detsember 2016].
- [17] D. Åse, „Creating a library website with login and multiple languages,“ 10 juuni 2016. [Võrgumaterjal]. <https://sparktutorials.github.io/2016/06/10/spark-basic-structure.html>. [Kasutatud 6. detsember 2016].
- [18] P. Webb, D. Syer, J. Long, S. Nicoll, R. Winch, A. Wilkinson, M. Overdijk, C. Dupuis ja S. Deleuze, „Spring Boot Reference Guide,“ [Võrgumaterjal].

- <http://docs.spring.io/spring-boot/docs/current/reference/html/>. [Kasutatud 6. detsember 2016].
- [19] Spring, „Guides,“ Spring, [Võrgumaterjal]. <https://spring.io/guides>. [Kasutatud 6. detsember 2016].
- [20] P. Webb, D. Syer, J. Long, S. Nicoll, R. Winch, A. Wilkinson, M. Overdijk, C. Dupuis ja S. Deleuze, „78. Security,“ [Võrgumaterjal]. <http://docs.spring.io/spring-boot/docs/current/reference/html/howto-security.html>. [Kasutatud 6. detsember 2016].
- [21] P. Webb, D. Syer, J. Long, S. Nicoll, R. Winch, A. Wilkinson, M. Overdijk, C. Dupuis ja S. Deleuze, „34. Sending email,“ [Võrgumaterjal]. <http://docs.spring.io/spring-boot/docs/current/reference/html/boot-features-email.html>. [Kasutatud 6. detsember 2016].
- [22] M. Pollack, T. Risberg ja O. Gierke, „1. Working with Spring Data Repositories,“ [Võrgumaterjal]. <http://docs.spring.io/spring-data/data-commons/docs/1.6.1.RELEASE/reference/html/repositories.html>. [Kasutatud 15. detsember 2016].
- [23] P. Webb, D. Syer, J. Long, S. Nicoll, R. Winch, A. Wilkinson, M. Overdijk, C. Dupuis ja S. Deleuze, „Part IV. Spring Boot features,“ [Võrgumaterjal]. <http://docs.spring.io/spring-boot/docs/current/reference/html/boot-features-external-config.html#boot-features-external-config-application-property-files>. [Kasutatud 6. detsember 2016].
- [24] P. Webb, D. Syer, J. Long, S. Nicoll, R. Winch, A. Wilkinson, M. Overdijk, C. Dupuis ja S. Deleuze, „Part IX. ‘How-to’ guides,“ [Võrgumaterjal]. <http://docs.spring.io/spring-boot/docs/current/reference/html/howto-spring-mvc.html#howto-use-thymeleaf-3>. [Kasutatud 6. detsember 2016].
- [25] PDFBox, „Apache PDFBox | A Java PDF Library,“ [Võrgumaterjal]. <http://pdfbox.apache.org/>. [Kasutatud 9. detsember 2016].
- [26] Flying Saucer, „flyingsaucerproject/flyingsaucer: XML/XHTML and CSS 2.1 renderer in pure Java,“ [Võrgumaterjal]. <https://github.com/flyingsaucerproject/flyingsaucer>. [Kasutatud 9. detsember 2016].

- [27] PDFBox, „Apache PDFBox | Examples,“ [Võrgumaterjal]. <http://pdfbox.apache.org/2.0/examples.html>. [Kasutatud 20. detsember 2016].
- [28] Flying Saucer, „Flying Saucer,“ [Võrgumaterjal]. <https://code.google.com/archive/p/flying-saucer/>. [Kasutatud 9. detsember 2016].
- [29] Sertifitseerimiskeskus, „Veebilehel isikutuvastus ID-kaardi ja digi-ID'ga > Kasutuse stsenaariumid > Arendajale,“ Sertifitseerimiskeskus, 1 juuni 2016. [Võrgumaterjal]. <http://www.id.ee/index.php?id=30268>. [Kasutatud 28. veebruar 2017].
- [30] M. Randala, M. Sõmermaa ja M. Paljak, „OpenID.ee teenuse tutvustus,“ OÜ Ideelabor, [Võrgumaterjal]. https://openid.ee/site_media/img/openid-teenuse-tutvustus.pdf. [Kasutatud 28. veebruar 2017].
- [31] „eID lühitutvustus,“ 26 jaanuar 2017. [Võrgumaterjal]. https://eid.eesti.ee/index.php/EID_1%C3%BCchitutvustus. [Kasutatud 28 veebruar 2017].
- [32] Baeldung, „X.509 Authentication in Spring Security | Baeldung,“ Baeldung, 12 oktoober 2016. [Võrgumaterjal]. <http://www.baeldung.com/x-509-authentication-in-spring-security>. [Kasutatud 28. veebruar 2017].
- [33] Oracle, „JavaMail API,“ [Võrgumaterjal]. <http://www.oracle.com/technetwork/java/javamail/index.html>. [Kasutatud 9. detsember 2016].
- [34] ZURB Foundation, „Foundation for Sites 6 Docs | Compatibility,“ [Võrgumaterjal]. <https://foundation.zurb.com/sites/docs/compatibility.html>. [Kasutatud 9. detsember 2016].
- [35] Bootstrap, „Getting started Bootstrap,“ [Võrgumaterjal]. <http://getbootstrap.com/getting-started/#support>. [Kasutatud 9. detsember 2016].
- [36] Bootstrap, „twbs/bootstrap: The most popular HTML, CSS, and JavaScript framework for developing responsive, mobile first projects on the web.,“ [Võrgumaterjal]. <https://github.com/twbs/bootstrap>. [Kasutatud 9. detsember 2016].
- [37] ZURB Foundation, „zurb/foundation-sites: The most advanced responsive front-end framework in the world. Quickly create prototypes and production code for sites that work

- on any kind of device.,” [Võrgumaterjal]. <https://github.com/zurb/foundation-sites>. [Kasutatud 9. detsember 2016].
- [38] Spring, „Getting Started · Validating Form Input,“ [Võrgumaterjal]. <https://spring.io/guides/gs/validating-form-input/>. [Kasutatud 12. detsember 2016].
- [39] B. Alex, L. Taylor, R. Winch ja G. Hillert, „2.1 What is Spring Security?,“ [Võrgumaterjal]. <http://docs.spring.io/spring-security/site/docs/4.2.1.BUILD-SNAPSHOT/reference/htmlsingle/#what-is-acegi-security>. [Kasutatud 12. detsember 2016].
- [40] B. Alex, L. Taylor, R. Winch ja G. Hillert, „5. Java Configuration,“ [Võrgumaterjal]. <http://docs.spring.io/spring-security/site/docs/current/reference/html/jc.html#jc-method>. [Kasutatud 12. detsember 2016].
- [41] B. Alex, L. Taylor, R. Winch ja G. Hillert, „23. Expression-Based Access Control,“ [Võrgumaterjal]. <http://docs.spring.io/spring-security/site/docs/current/reference/html/el-access.html#access-control-using-preauthorize-and-postauthorize>. [Kasutatud 12. detsember 2016].
- [42] Thymeleaf, „Tutorial: Using Thymeleaf,“ 19 november 2016. [Võrgumaterjal]. <http://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html#using-thtext-and-externalizing-text>. [Kasutatud 15. detsember 2016].
- [43] The PostgreSQL Global Development Group, „PostgreSQL: About,“ [Võrgumaterjal]. <https://www.postgresql.org/about/>. [Kasutatud 9. detsember 2016].
- [44] The PostgreSQL Global Development Group, „PostgreSQL 9.6.1 Documentation,“ [Võrgumaterjal]. <https://www.postgresql.org/files/documentation/pdf/9.6/postgresql-9.6-US.pdf>. [Kasutatud 9. detsember 2016].
- [45] B. Pugh, A. Loskutov ja K. Lea, „FindBugs Bug Descriptions,“ The University of Maryland, 2015. [Võrgumaterjal]. http://findbugs.sourceforge.net/bugDescriptions.html#EI_EXPOSE_REP. [Kasutatud 22. jaanuar 2017].
- [46] Natera, „Panorama International Requisition,“ 2017.

[47] Harmony, „Prenatal Test,“ 2017.

Lisad

I Natera NIPT tellimise vorm



How to Complete Requisition Forms to Avoid Test Delays

Natera is unable to complete reporting of test results when the following fields are missing from the req form. Please ensure these fields are complete for all orders.

panorama
natera prenatal screen

PANORAMA INTERNATIONAL REQUISITION

PLEASE COMPLETE ALL FIELDS IN ENGLISH. REQUISITION FORMS SUBMITTED WITH MISSING INFORMATION MAY CAUSE A DELAY IN TURNAROUND TIME OF THE TEST.

1 PATIENT INFORMATION

1 First Name
2 Last Name
3 Date of Birth

Address
City
Country
Telephone
Email

Weight (kg) Height (cm)

Is patient pregnant? 1st Trimester 2nd Trimester 3rd Trimester

4 Expected Due Date **! Patient must be at least 9 weeks gestation.**

Patient must be at least 9 weeks gestational age.

Is this a twin, vanished twin, surrogate, or egg donor case? Yes No
If Yes, DO NOT submit a blood sample. We do not accept these cases.

2 CLINIC INFORMATION

5 Clinic Name
Telephone
Natera® LIMS ID

6 Ordering Clinician Name

3 SCREENING OPTIONS

! Clearly select the test. The options you see on your own form may differ.

7 Date of Sample Collection **! Sample must be received within 7 days of draw.**

8 SELECT SCREENING OPTION BELOW

PANORAMA PRENATAL PANEL Chromosomes 13, 18, 21, X & Y; Triploidy

PANORAMA PRENATAL PANEL + 22q11.2 DELETION Chromosomes 13, 18, 21, X & Y; Triploidy; 2;

PANORAMA EXTENDED PANEL Chromosomes 13, 18, 21, X & Y; Triploidy; 22q11.2 deletion PLUS

Check to add fetal sex to report (available with any screening option above)

4 DISPOSITION OR RETENTION OF SAMPLES


Laboratory (Reseller) represents and confirms that the patient has given informed consent in compliance following sample disposition or retention policy: PATIENT UNDERSTANDS AND CONSENTS THAT: (i) her the United States for performance of the test; (ii) Natera may retain the patient's leftover, de-identified sample technology advancement, research & development, product validation and quality assurance, independent third-party partners, either in or outside the United States; and (iii) patient and patient's heirs will not receive rights to any resulting products or discoveries.

REQUIRED FIELDS

- 1 Patient first name
- 2 Patient last name
- 3 Patient date of birth
- 4 Patient expected due date
- 5 Clinic name
- 6 Clinician full name
- 7 Sample collection date
- 8 Test selection

Joonis 6. Natera NIPT geenitesti tellimise vorm [46]. Firmalt Natera NIPT testi tellimiseks tuleb joonisel olev vorm täita ja saata Naterale koos patsiendi vereprooviga.

II Harmony NIPT tellimise vorm

Ordering Lab Specimen ID		Place the -PAT barcode label here
1 Patient Information		
First Name	Account number 14389	Clinic Name Tartu University Hospital
Last Name	Ordering Clinician	
Street Address	Street Address	
City	State/Province	City
Country	ZIP/Postal Code	Country
Phone		Fax
Medical Record Number	Gender <input type="checkbox"/> Female <input type="checkbox"/> Male	Referring Clinician (Copy of Results)
Weight (kg)	Height (cm)	Referring Clinician Fax
3 Patient Informed Consent		
<p>My signature on this form indicates that I have read, or had read to me, the informed consent on the back of this form, and I understand it. I have had the opportunity to ask questions and discuss the test, including the purposes and possible risks, with my healthcare provider or someone my healthcare provider has designated. I know that I may obtain professional genetic counseling if I wish, before signing this consent. I give permission to Ariosa Diagnostics to perform the Harmony Prenatal Test.</p> <p><input type="checkbox"/> Check to opt-out of laboratory studies and the patient registry (described on page 2)</p> <p><small>Details on Ariosa's policies and procedures governing patient privacy and health information, including patient rights regarding such information, can be found at www.ariosadx.com/patient</small></p> <p>Patient Signature _____</p>		
4 Clinician Signature		
<p>I attest that this patient has been informed about details of the test and its capabilities and limitations, and has given consent for this test.</p> <p>Clinician Signature _____</p>		
5 Required Test Information		
<input checked="" type="checkbox"/> Harmony Prenatal Test <input type="checkbox"/> with Y Analysis* <input type="checkbox"/> with X,Y Analysis* <small>* Singletons only</small>		
Draw Date	Is this a redraw? <input type="checkbox"/> Yes <input type="checkbox"/> No	Mother's Birthdate
Gestational Age	Measured on	by: <input type="checkbox"/> U/S <input type="checkbox"/> LMP <input type="checkbox"/> IVF
LMP Date	EDD/EDC Date	by: <input type="checkbox"/> U/S <input type="checkbox"/> LMP <input type="checkbox"/> IVF <input type="checkbox"/> PE
# of Fetuses <input type="checkbox"/> 1 <input type="checkbox"/> 2	IVF Pregnancy <input type="checkbox"/> Yes <input type="checkbox"/> No	Egg Donor is <input type="checkbox"/> Self <input type="checkbox"/> Non-self
Clinical Indications		Age at retrieval: _____ years
<input type="checkbox"/> Advanced maternal age (>35) (V23.81-23.82) <input type="checkbox"/> Ultrasound abnormality (V28.3) <input type="checkbox"/> Other genetic screening (V82.79) <input type="checkbox"/> Positive maternal serum screen (796.5) <input type="checkbox"/> Other antenatal screening (V28.89) <input type="checkbox"/> Other (specify): _____		
6 Billing Information		
<input type="checkbox"/> Bill Clinic <input type="checkbox"/> Bill Patient		

Joonis 7. Harmony NIPT geenitesti tellimise vorm [47]. Firmalt Harmony NIPT testi tellimiseks tuleb joonisel olev vorm täita ja firmale Harmony saata koos patsiendi vereprooviga.

III Kasutuslugu patsiendi ülevaatusse lisamisest

Kasutusloos kasutaja täidab ja postitab ülevaatusvormi. Tabelis 4 on esitatud tavaline voog ja tabelis 5 alternatiivne voog, kui kasutaja eksib tabelis 4 tegevuse number 2 vastu.

Põhitegutseja: arst

Kaetud funktsionaalsed nõuded: PÜ3–PÜ9

Eeldused:

- Kasutaja on sisse loginud ja paikneb pealehel.
- Kasutajal on õigus patsiendi ülevaatusse andmete sisestamiseks.
- Süsteem on eestikeelne.

Tabel 4. Patsiendi ülevaatusse lisamise tavaline voog.

Tegevuse järjekord	Tegevus
1	Kasutaja vajutab tiitelribal väljale „Sisesta ülevaatus“.
2	Kasutaja täidab väljad vastavalt tabel 1 korrektsele sisendile.
3	Kasutaja vajutab nupule „Kinnita andmed“.
4	Andmed postitakse andmebaasi.
5	Kasutaja suunatakse samale lehele, kuid lahtrid on tühjendatud.
6	Lehel on teade „Edukalt salvestatud!“.

Tabel 5. Patsiendi ülevaatusse lisamise alternatiivne voog.

Tegevuse järjekord	Tegevus
2. A	Kasutaja sisestab mõne välja eksides tabelis 1 nõutud korrektse sisendi vastu.
3. A	Kasutaja vajutab nupule „Kinnita andmed“.
4. A	Kasutajale kuvatakse sama leht, kus vigane sisend on eemaldatud ja vigased kohad on markeeritud.
5. A	Kasutaja parandab vead ja vajutab nupule „Kinnita andmed“.
6. A	Kasutaja suunatakse samale lehele, kus lahtrid on tühjendatud.
7. A	Lehel on teade „Edukalt salvestatud!“.

IV Kasutuslugu raporti loomisest

Kasutajaloos eksperthindaja kinnitab analüüsi tulemuse, mille tulemusel valmib raport. Tabelis 6 on esitatud tavaline kasutusvoog.

Põhitegutseja: eksperthindaja

Kaetud funktsionaalsed nõuded: RL2–RL6

Eeldused:

- Kasutaja on sisse loginud ja paikneb pealehel.
- Kasutajal on süsteemis õigus raporti loomiseks.
- Kasutajal on teada patsiendi isikukood ja proovinumber, mille kohta ta hinnangu annab.
- Patsiendi vereproovi tulemused on andmebaasis olemas.
- Süsteem on eestikeelne.

Tabel 6. Raporti loomise tavaline voog.

Tegevuse järjekord	Tegevus
1	Kasutaja valib tiitelribal valiku „Raport“.
2	Kasutaja sisestab patsiendi isikukoodi.
3	Kasutaja vajutab nupule „Päri andmed“.
4	Kasutaja valib „Vali ülevaatus“ alt sobiva ülevaatus.
5	Kasutaja vajutab nupule „Hinda“.
6	Kasutajale kuvatakse punktis RL3 esitatud informatsioon
7	Kasutaja täidab vastavalt andmetele hinnangu lahtri ja valib otsuse.
8	Kasutaja vajutab nupule „Kinnita ja teavita tulemustest“.
9	Süsteem salvestab raporti ja teavitab ülevaatusel sisestatud e-posti aadressile tulemuste valmimisest.
10	Süsteem suunab kasutaja pealehele.
11	Pealehel on teade „Edukalt salvestatud!“.

V Kasutuslugu raporti alla laadimisest

Kasutusloos arst laeb alla geenitesti tulemuste raporti. Tabel 7 esitab tavalise ja alternatiivse voo.

Põhitegutseja: arst

Kaetud funktsionaalsed nõuded: RV2–RV4

Eeldused:

- Kasutaja on sisse loginud ja paikneb pealehel.
- Kasutajal on süsteemis õigus raporti vaatamiseks.
- Kasutajal on teada patsiendi isikukood.
- Kasutajal on teada vereproovi kood.
- Eksperthindaja on raporti loonud.
- Süsteem on eestikeelne.

Tabel 7. Raporti alla laadimise tavaline voog.

Tegevuse järjekord	Tegevus
1	Kasutaja valib tiitelribal valiku „Raport“.
2	Kasutaja sisestab patsiendi isikukoodi.
3	Kasutaja vajutab nupule „Päri andmed“.
3. A	Kui patsiendi kohta informatsioon puudub, siis kuvatakse kasutajale tekst: „Patsienti ei leitud“ ja voo täitmine lõppeb.
4	Kasutaja valib rippmenüüst „Vali ülevaatus“ sobiva ülevaatus.
4. A	Kui eksperthindaja pole patsiendi kohta raportit loonud, siis on ülevaatusete lahter tühi ja voo täitmine lõppeb.
5	Kasutaja vajutab nupule „Näita raportit“.
6	Kasutajale kuvatakse raport, kus on kirjas RV3 välja toodud informatsioon
7	Kasutaja vajutab nupule „Laadi alla“.
8	Raport laaditakse PDF-kujul kasutaja arvutisse alla.

VI Kasutuslugu kasutaja lisamisest

Kasutusloos süsteemi kasutaja X soovib luua uue kasutaja inimesele Y. Kasutusloo voog on välja esitatud tabelis 8.

Põhitegutseja: kasutaja

Kaetud funktsionaalsed nõuded: KL2–KL3

Eeldused:

- Kasutaja on sisse loginud ja paikneb pealehel.
- Kasutajal on süsteemis õigus kasutajate lisamiseks.
- Süsteem on eestikeelne.

Tabel 8. Kasutaja lisamise tavaline voog.

Tegevuse järjekord	Tegevus
1	Kasutaja X valib tiitelribal valiku „Lisa kasutaja“.
2	Kasutaja X täidab vormi: <ul style="list-style-type: none">• X sisestab uue kasutaja eesnime.• X sisestab uue kasutaja perenime.• X sisestab uue kasutaja isikukoodi.• X sisestab uue kasutaja e-posti aadressi.• X määrab uue kasutaja õigused nii, et need on samad, mis kasutajal X endal.
3	Kasutaja X vajutab nupule „Loo kasutaja ja teavita teda e-post teel“.
3. A	Kui kasutaja X eksis mõne välja piirangu vastu punktis 2, siis kuvatakse kasutajale X sama lehekülge, kus on vigane sisend eemaldatud ja sisendi kast ära markeeritud. Voog jätkub punktist 2.
4	Süsteemi luuakse uus kasutaja.
5	Kasutajale X kuvatakse teade „Edukalt salvestatud!“.
6	Kasutajale Y saadetakse e-post, kus on parool ja kasutajatunnus, millega ta saab süsteemi siseneda.

VII Kasutuslugu parooli muutmisest

Kasutusloos kasutaja soovib vahetada iseenda parooli. Tabelis 9 on esitatud kasutusvoog.

Põhitegutseja: kasutaja

Kaetud funktsionaalsed nõuded: ÜN2

Eeldused:

- Kasutaja on sisse loginud ja paikneb pealehel.
- Kasutajal on aktiveeritud JavaScript brauseris.
- Süsteem on eestikeelne.

Tabel 9. Parooli muutmise tavaline voog.

Tegevuse järjekord	Tegevus
1	Kasutaja valib tiitelribal valiku „Sätted“.
2	Kasutaja kirjutab vana parooli lahtrisse „Vana parool“.
3	Kasutaja kirjutab vähemalt 5 tähemärgist koosneva uue parooli lahtrisse „Uus parool“.
4	Kasutaja kirjutab punktis 3 kirjutatud parooli lahtrisse „Korda parooli“
4. A	Sisestatud parool ei kattu punktis 4 sisestatud parooliga. Punktis 5 mainitud tiitel ei muutu roheliseks ja nupp „Vaheta parooli“ ei ole vajutatav. Kasutaja alustab punktist 3 uuesti.
5	„Korda parooli“ lahtri tiitel värvub roheliseks.
6	Nupp „Vaheta parooli“ muutub vajutatavaks.
7	Kasutaja vajutab nupule „Vaheta parooli“.
8	Kasutaja parool uuendatakse.

VIII Kasutuslugu alluva õiguste muutmisest

Kasutusloos kasutaja X võtab kasutajalt X ära õiguse raporti vaatamiseks. Kasutusloo voog on esitatud tabelis 10.

Põhitegutseja: kasutaja X

Kaetud funktsionaalsed nõuded: ÜN3

Eeldused:

- Kasutaja X on sisse loginud ja paikneb pealehel.
- Kasutaja X on varasemalt kasutaja Y loonud.
- Kasutaja X teab kasutaja Y andmeid.
- Kasutajal X on õigus raporti vaatamiseks.
- Kasutajal Y on õigus raporti vaatamiseks.
- Süsteem on eestikeelne.

Tabel 10. Alluva õiguste muutmise tavaline voog.

Tegevuse järjekord	Tegevus
1	Kasutaja X valib tiitelribal valiku „Halda juurdepääsu“.
2	Kasutaja X eemaldab valiku „Vaata raporteid“ kasutaja Y realt.
3	Kasutaja X vajutab nuppu „Uuenda“.
4	Kasutaja Y õigused uuendatakse.

IX Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Priit Paluoja

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Andmebaas ja selle veebipõhine kasutajaliides loote sünnieelse sõeluuringu kliinilistele proovidele“,

mille juhendajad on Kaarel Krjutškov ja Priit Adler,

1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **23.04.2017**