

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Riho Kalda 202024IADB

Geodeetiliste punktide andmekogu turvalise veebirakenduse arendus

Bakalaureusetöö

Juhendaja: Aleksei Talisainen
MSc

Tallinn 2022

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Riho Kalda

15.05.2022

Annotatsioon

Käesoleva bakalaureusetöö eesmärgiks on luua uus, turvaline, geodeetiliste punktide andmekogu rakendus.

Analüüsi käigus uuritakse erinevaid tehnoloogiaid ning kaalutakse nende sobivust loodava rakenduse jaoks. Rõhku pannakse turvalisusele ja tuuakse välja võimalikud haavatavused ning meetmed nende likvideerimiseks.

Bakalaureusetöö käigus luuakse ees- ja tagarakendus, võttes arvesse analüüsi käigus tehtud otsuseid. Rakendus võimaldab sooritada otsinguid geodeetiliste punktide andmekogusse, vaadata detailset infot punkti kohta ning näitab punkti asukohta kaardil. Lisaks on võimalik näha ka pildimaterjali, mis on punkti kohta olemas.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 38 leheküljel, 6 peatükki, 17 joonist, 2 tabelit.

Abstract

Development of Secure Geodetic Point Database Web Application

The goal of this thesis is to create a new and secure Geodetic Point Database web application.

In the analysis section author examines different technologies and considers their suitability for the new application. Special attention goes to security. Potential vulnerabilities are being defined and remediation against them is described.

As a result, frontend and backend applications will be created, considered the decisions made during analysis section. The application allows users to perform queries into the Geodetic point database, display detailed information about the geodetic point. Also, the location is shown on the map. In addition, it is possible to browse pictures of the geodetic point.

The thesis is in Estonian and contains 38 pages of text, 6 chapters, 17 figures, 2 tables.

Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> , tarkvaraliidese tüüp, mis osutab teenust mõnele teisele tarkvarale
ASVS	<i>Application Security Verification Standard</i> , Rakenduse turvalisuse kontrollimise standard. Projekti eesmärk on luua avatud turvalisuse standard veebirakenduste ja -teenuste jaoks
AWT	<i>Abstract Window Toolkit</i> , Java API graafiliste liideste loomiseks
Base64	Kahendkood-tekstiks muundamise algoritm
BK77	Balti 1977. aasta kõrgussüsteem. Kroonlinna null
CERT-EE	Organisatsioon Eestis, mis käsitleb küberintsidente
CBA	<i>Component Based Architecture</i> , arhitektuurimuster, milles tarkvara on loodud iseseisvatest komponentidest.
Clickjacking	Klõpsamine (kombinatsioon sõnadest klõps ja kaaperdamine). Rünne, mille käigus kasutaja manipuleeritakse klõpsama mõnel teisel elemendil, kui see, mis on nähtaval
Cross-site scripting	Murdskriptimine. Rünne, mille käigus saadetakse andmeteks maskeeritud kood, mida rakenduses käsitletakse kui koodi
Command Injection	Käsu süstimine. Rünne, mille käigus saadakse käivitada rakendusega samas süsteemis operatsioonisüsteemi tasemel käske
CSS	<i>Cascading Style Sheets</i> , kirjeldab veebilehtede esitust (kirjastiil, värvid jne)
CSV	<i>Comma-separated values</i> , tekstifaili tüüp, milles kasutatakse koma väärtuste eraldamiseks
DOM	<i>Document Object Model</i> , liides esitamaks dokumenti loogilise puuna
EH2000	Euroopa kõrgussüsteem. Amsterdami null
Ellipsoid	Maa tõelisele kujule lähedane pöördellipsoid
EST-GEOID2017	Eesti geoidi mudel
exe	Käivitatav failitüüp Windows operatsioonisüsteemis
Geoid	Maa raskusjõuvalja ekvipotentsiaalpind, mis ühtib merede ja ookeanide "häirimata" veepinnaga [1]
GIS	Geograafiline infosüsteem

HTML	<i>HyperText Markup Language</i> , keel dokumentide kujundamiseks ning veebibrauseis näitamiseks
Jar	<i>Java archive</i> , pakitud faili vorming, mis sisaldab java klasse, sellega seotud metainfot ning ressursifaile
JPQL	<i>Java Persistence Query Language</i> , päringukeel, mis võimaldab andmebaasipäringuid teha rakenduse olemite mudeli põhjal
JPA	<i>Java Persistence API</i> , Java spetsifikatsioon objekti-suhte kaardistuse(ORM) tööriistadele
JPEG	<i>Joint Photographic Experts Group</i> , piltide vorming, kasutatakse kadudega pakkimist
JSON	<i>JavaScript Object Notation</i> , tekstipõhine vorming struktureeritud andmete näitamiseks kasutades JavaScripti objekti süntaksit
JSX	<i>Javascript XML(eXtensible Markup Language)</i> , Reacti laiendus lihtsustamaks HTML-i ja JavaScripti koos kirjutamist.
KeM	Keskkonnaministeerium
KeMIT	Keskkonnaministeeriumi Infotehnoloogiakeskus
L-EST97	Eesti tasapinnaliste ristkoordinaatide süsteem
MVC	<i>Model-View-Controller</i> , tarkvaradisaini muster, mis jagab rakenduse kolmeks loogiliseks komponendiks: mudel, vaade, kontrollor.
OWASP	<i>Open Source Foundation for Application Security</i> , mittetulunduslik organisatsioon, mille eesmärgiks on parendada tarkvara turvalisust
PDF	<i>Portable Document Format</i> , platvormist sõltumatu vorming dokumentide kuvamiseks
PHP	<i>Hypertext Preprocessor</i> , populaarne veebiarenduses kasutatav skriptimiskeel
POST	Päringu meetod, mida toetab HTTP (Hypertext Transfer Protocol)
PostGIS	PostgreSQL andmebaasisüsteemi laiendus, lisab ruumiandmete toe
REST	<i>Representational State Transfer</i> , tarkvaraarhitektuuriline stiil, mis loodi juhiseks veebiteenuste loomiseks
RESTful	API, mis on loodud REST arhitektuurilise stiili juhiseid järgides
SQL	<i>Structured Query Language</i> , keel info pärimiseks ja haldamiseks relatsioonilises andmebaasis
SQL Injection	SQL süstimine. Rünne, mille käigus süstitakse koodi SQL päringusse

TypeScript	Tugevalt ja staatiliselt tüübitud programmeerimiskeel
war	<i>Web Archive</i> , vorming Java veebirakenduste pakkimiseks.
WFS	<i>Web Feature Service</i> , Standard geoinfo vektorformaadis jagamiseks üle HTTP protokoll
WGS84	<i>World Geodetic System 1984</i> , Ülemaailmne geodeetiline süsteem
WMS	<i>Web Map Service</i> , Teenus kaardifailide üle HTTP protokoll jagamiseks.
WSTG	<i>Web Security Testing Guide</i> , juhend veebirakenduste turvalisuse testimiseks

Sisukord

1 Sissejuhatus	12
2 Probleemi püstitus	13
2.1 Eesmärk	14
2.2 Autori roll	14
3 Lähtetingimused	15
4 Analüüs.....	18
4.1 Tehnoloogia valik.....	18
4.2 Tagarakenduse raamistiku valik	18
4.2.1 Spring Boot.....	18
4.2.2 Django	19
4.2.3 Laravel.....	20
4.2.4 Sobivaima raamistiku valik	21
4.3 Eesrakenduse raamistiku valik	22
4.3.1 Angular	22
4.3.2 React.js	23
4.3.3 Sobivaima raamistiku valik	24
4.4 REST arhitektuuriline stiil.....	25
4.5 Tarkvaraarhitektuuri mustrid	26
4.5.1 Kihiline arhitektuur	26
4.5.2 Mikroteenuste arhitektuuri muster	27
4.5.3 Sobivaima mustri valik.....	27
4.6 Rakenduste turvalisuse kontrollimise standard	28
4.6.1 Kuidas ASVS rakendada	28
4.7 Veebiturbe testimise juhend	30
4.8 Kaardimooduli teegi valik	31
4.8.1 Leaflet.....	31
4.8.2 OpenLayers.....	32
4.8.3 X-GIS2 kaardirakendus	32
4.8.4 Kaardikihid	32

4.8.5 Sobivaima teegi valik	33
4.9 Kalkulaatorid	34
5 Lahenduse teostus	36
5.1 Tagarakendus	36
5.2 Eesrakendus	40
5.3 Turvalisus	44
5.3.1 Kasutaja sisend	44
5.3.2 Piltide allalaadimine	46
5.4 Kaardimoodul	46
5.5 Kalkulaatorid	48
6 Kokkuvõte	49
Kasutatud kirjandus	50
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	55
Lisa 2 – Kohalike koordinaadisüsteemide loend	56

Jooniste loetelu

Joonis 1. Kihilise arhitektuuri skeem [38].....	26
Joonis 2. Mikroteenuste arhitektuuri skeem [38]	27
Joonis 3. X-GIS2 probleemi näide	33
Joonis 4. OpenLayers teegis EPSG:3301 projektsiooni defineerimine. [55]	34
Joonis 5. Leaflet teegis EPSG:3301 projektsiooni defineerimine. [55]	34
Joonis 6. GPA tagarakenduse skeem, ilma kalkulaatorite API lõpp-punktideta.....	36
Joonis 7. Koodinäide API kaudu pildi saatmisest	37
Joonis 8. Kvaliteedi võrdlus. Vasakul teisendatud JPG, paremal originaal PDF.....	38
Joonis 9. Kalkulaatorite API skeem	39
Joonis 10. Sisseloetud baidijadast baitide lugemine ja 32-bitise ujukomaarvu loomine	40
Joonis 11. Utiliidi kaustateed ning käsurea argumendid	40
Joonis 12. Eesrakenduse voodiagramm.....	41
Joonis 13. Nime järgi otsingu vorm	42
Joonis 14. Rakenduses piltide kuvamise näide.....	43
Joonis 15. CSV failist kirjete laadimine ja loodud vastuse link	44
Joonis 16. Poolikud märgistused kaardil	46
Joonis 17. Kihi lisamine ühe ruuduna	47

Tabelite loetelu

Tabel 1. Spring Boot ja Laravel võrdlus.....	21
Tabel 2. Angular ja React võrdlus	24

1 Sissejuhatus

Geodeetiliste punktide andmekogu asutamine ja andmekogu pidamise põhimäärus on kehtestatud Vabariigi Valitsuse 22.12.2011 määrusega nr 164 "Geodeetiliste punktide andmekogu asutamine ja andmekogu pidamise põhimäärus". Määrus on kehtestatud "Ruumiandmete seaduse" § 36 lõike 2 alusel. Määrusega on andmekogu vastutavaks töötajaks määratud Maa-amet. Lisaks on määruses ära toodud ka andmed, mida registrisse peab kandma. [2], [3], [4]

Ruumiandmete seaduse järgi on geodeetiliseks punktiks kohtkindlalt paigaldatud geodeetilise märgi asukoht maapinnal, pinnases või ehitisel, millele on arvutatud koordinaadid ning vajaduse korral arvutatud kõrgused ja määratud raskuskiirenduse väärtus. [4] Geodeetiliste punktide koordinaatide abil saab looduses määrata mistahes objekti asukohta ning need on abiks topograafiliste kaartide koostamisel, ehituseelsete tööde tegemisel, kinnistu piiride täpsustamisel jne. [2]

Geodeetilisi märke hakati tähistama juba üle sajandi tagasi ning vanim kasutuses olev märk on paigaldatud aastal 1891 Kuremäe kloostri juurde. [2]

Käesoleva bakalaureusetöö käigus loob autor Maa-ametile uue rakenduse, läbi mille saab teostada päringuid geodeetiliste punktide andmekogu andmetele ligipääsemiseks. Autor leiab tehnilise analüüsi käigus rakenduse loomiseks sobivaimad tehnoloogiad ja tuvastab võimalikud ründevektorid rakenduse vastu ning leiab lahendused nende haavatavuste kõrvaldamiseks. Autor loob kalkulaatorite kasutamiseks RESTful API, mis järgib REST arhitektuuri soovitusi. Võimaldatakse teha nii ühe kirje kui ka massarvutusi.

Bakalaureusetöö skooپی ei jää andmebaasisüsteemi valik. Kasutusel on PostgreSQL andmebaas koos paigaldatud PostGIS laiendusega.

Autor kirjeldab lahenduse teostust ning toob välja ette tulnud probleemid ja probleemide kõrvaldamiseks leitud lahendused.

2 Probleemi püstitus

Maa-amet on teinud geodeetiliste punktide andmekogu ka avalikkusele kättesaadavaks. Tegemist on ühe veebirakendusega, mis on loodud PHP-s. CERT-EE sooritas rakendusele turvatestid, mille käigus leiti tõsiseid haavatavusi. Kriitiliste vigade hulgas leiti *Cross Site Scripting* ja *Command Injection* nõrkused. Probleemide hulgas oli ka see, et failide üleslaadimisel ei kontrollitud seda, mis failiga tegemist on ja nende suurus oli piiramata. Kõigele lisaks puudus ka koodil igasugune struktuur. Kõik oli segamini - HTML, loogika ja SQL laused.

Lisaks on rakenduse juures ka kalkulaatorid, mis on mõeldud erinevate koordinaatide ümberarvutamiseks L-EST97 koordinaatide süsteemi. [5]

BK77 ja EH2000 kõrguste ülemineku kalkulaator, mis võimaldab BK77 kõrguste ümberarvutamist EH2000 kõrgusteks ja vastupidi. [5]

EST-GEOID2017 kalkulaator, mida kasutatakse EUREF-EST97 ellipsoidaalsete kõrguste ümberarvutamisel EH2000 kõrgusteks ja vastupidi. [5]

Koordinaatide kalkulaator, mille abil on võimalik teostada ümberarvutusi L-EST97 koordinaadisüsteemist WGS84 koordinaadisüsteemi. [5]

Globaalne geodeesia kalkulaator, mis on mõeldud koordinaatide teisendamiseks ühest koordinaatide süsteemist teise. [5]

Kohalik geodeesia kalkulaator, mis on mõeldud kohalike koordinaatide teisendamiseks L-EST97 koordinaadisüsteemi. [5]

2.1 Eesmärk

Kuna vana rakenduse kood on halva kvaliteediga, siis on otstarbekas luua uus rakendus. Uus rakendus peaks võimaldama sooritada samu tegevusi, mida vana rakendus, kuid on oluline, et tulemus oleks ka turvaline.

Analüüsitakse erinevaid vahendeid, millega uus rakendus luua. Defineeritakse rakenduse juures olevad potentsiaalsed turvanõrkused.

Kalkulaatorite kasutamisel peab olema võimalus teha arvutusi nii ühe kaupa kui ka massarvutusi.

Praktilise osa käigus luuakse rakendus kasutades analüüsi käigus leitud tehnoloogiaid.

2.2 Autori roll

Autor ei ole otseselt seotud Maa-ametiga, kelle jaoks rakendus luuakse. Maa-ameti poolseks kontaktiks on Maa-ameti Geodeesia osakonna juhataja, kellega on võimalus vajadusel konsulteerida.

3 Lähtetingimused

Antud peatükis kirjeldab autor uue rakenduse funktsionaalseid nõudeid. Nõuded on autor saanud analüüsidest olemasoleva rakenduse lähtekoodi.

- Rakendus peab võimaldama teostada otsingut andmebaasis oleva unikaalse identifitseerimisnumbri järgi. Leitud identifitseerimisnumbriga geodeetilise punkti kohta näidatakse detailset infot.
- Rakendus peab võimaldama teostada otsingut geodeetilise punkti nime järgi. Leitakse geodeetilised punktid, mille nimi sisaldab otsitavat sõnet. Leitud punktid kuvatakse rakenduses tabelis.
- Rakendus peab võimaldama teostada otsingut koodnumbri järgi. Koodnumber koosneb kolmest arvust, koodnumbri vorming on xxx-xxx-xxx. Leitud punktid kuvatakse rakenduses tabelis.
- Rakendus peab võimaldama otsida etteantud punkti ümber etteantud raadiuses olevad teised geodeetilised punktid. Sisestada saab nii L-EST97 vormingus kui ka geograafilisi koordinaate (geograafiline laius ja pikkus), raadius meetrites. Tulemused kuvatakse tabelis.
- Rakendus peab võimaldama näidata detailvaates geodeetilise punkti asukohta kaardil.
- Rakendus peab leidma andmebaasist geodeetilise punktiga seotud pildimaterjalide ID-d ning leitud pilte rakenduses geodeetilise punkti info juures näitama. Pildimaterjal võib olla nii JPEG kui ka PDF formaadis.
- Rakendus peab võimaldama teha arvutusi EUREF-EST97 ellipsoidaalsete kõrguste ümberarvutamiseks EH2000 kõrgusteks ja vastupidi: EUREF-EST97 ellipsoidaalsed kõrgused – normaalkõrgused EH2000 kalkulaator (EST-GEOID2017).

- Rakendus peab võimaldama teha ümberarvutusi L-EST97 koordinaadisüsteemist WGS84 koordinaadisüsteemi: geodeetilised koordinaadid EUREF-EST97 – tasapinnalised ristkoordinaadid L-EST97 kalkulaator.
- Rakendus peab võimaldama teha koordinaatide teisendamist ühest koordinaatide süsteemist teise: globaalne translaator. Koordinaadisüsteemid, mida toetama peab:
 - 1963.a. 0. tsoon;
 - 1942.a. telgmeridiaan 24;
 - 1963.a. 1. tsoon;
 - 1963.a. 2. tsoon;
 - 1942.a. 4. tsoon;
 - 1942.a. 5. tsoon;
 - Baaskaart;
 - EUREF-EST97, WGS-84 geodeetiline;
 - Läti TM;
 - L-EST97;
 - Pulkovo80 geodeetiline;
 - EUREF-EST97 Gauss-Krüger 21;
 - EUREF-EST97 Gauss-Krüger 24;
 - EUREF-EST97 Gauss-Krüger 27;
 - EUREF-EST97 UTM34;
 - EUREF-EST97 UTM35.

- Rakendus peab võimaldama teha kohalike koordinaatide arvutamist L-EST97 koordinaadisüsteemi ja vastupidi (Lisa 2): kohalik translaator.
- Rakendus peab võimaldama BK77 kõrguste ümberarvutamist EH2000 kõrgusteks: üleminekumudel BK77 -> EH2000.

4 Analüüs

Antud peatükis analüüsib autor erinevaid tehnoloogiaid ning valib rakenduse loomiseks parimad.

4.1 Tehnoloogia valik

Keskkonnaministeeriumi Infotehnoloogiakeskuses 18.11.2021 jõustunud dokumendiga „KeM haldusala tehnoloogiline profiil“ määratakse ära tehnoloogiad, programmeerimiskeeled, raamistikud ning muud komponendid, mille arendamise ja haldamise kogemus on asutusel olemas. [6] Rakenduse loomisel tuleb võtta see dokument tehnoloogiate valikul aluseks. Selle dokumendi põhjal on tagarakenduse tegemiseks soovitatavad raamistikud Javas Spring Boot, Pythonis Django, PHP-s Laravel, Symphony ning Phalcon. Eesrakenduse tegemiseks on soovitatavad Angular raamistik ning React.js teek. [6]

4.2 Tagarakenduse raamistiku valik

Tagarakendus on veebirakenduse osa, mis töötab serveri pool ja vahendab kliendile andmeid, vajadusel muutes andmed nõutud kujule. Kasutaja reeglina tagarakendusega otse ei suhtle, vaid teeb seda läbi eesrakenduse. [7]

Käesoleva bakalaureusetöö käigus võrdleb autor omavahel Spring Boot, Django ning Laravel raamistikke. Kuna Laravel on populaarseim PHP raamistik, siis jätab autor teised KeM haldusala tehnoloogilises profiilis ära nimetatud PHP raamistikud kõrvale. [8]

4.2.1 Spring Boot

Spring Boot on Pivotal Team'i arendatav avatud lähtekoodiga Spring raamistiku moodul. Sellega saab luua iseseisvaid rakendusi minimaalse Spring'i konfigureerimisega. Palju keerukust on arendaja eest ära peidetud, mis tähendab, et arendajal on kerge alustada ning jõuab lihtsamalt tulemuseni. See raamistik sobib eriti hästi lihtsate Spring'il baseeruvate RESTful teenuste loomiseks. [9]

Spring Boot'i positiivsed aspektid:

- Lihtne seadistada ja hallata, võimalusel konfigureeritakse Spring'i ja kolmanda osapoole teeke automaatselt. [10]
- Sisseehitatud Tomcat, Jetty või Undertow veebiserver, mis tähendab, et rakenduse juurutamisel pole vaja eraldi veebiserverit paigaldada ning pole vaja opereerida .war failidega. [10]
- Sobib väikeste teenuste loomiseks. [11]

Negatiivsed aspektid [11]:

- Vähene kontroll. Spring Boot loob palju sõltuvusi, mida ei kasutata ning mille tulemusel tekib suur juurutusfail.
- Paljude arendajate väitel pole sobiv suurte projektide jaoks.

4.2.2 Django

Django on Pythonis kirjutatud vabavaraline serveripoolne raamistik veebirakenduste loomiseks. Django on loodud eesrakenduste arendajaid silmas pidades. Django mallide keel on disainitud olemaks mugav ja kergesti õpitav kõigile, kes on HTML-iga harjunud töötama. [12]

Üheks Django põhieesmärgiks on lihtsustada arendajate tööd. Võimaldatakse kiiret arendust ning järgitakse „Ära korda ennast“ filosoofiat, mis tähendab seda, et kasutajad saavad olemasolevat koodi korduvkasutada. [13]

Positiivsed aspektid:

- Django raamistik sisaldab peaaegu kõike, mida arendajal vaja võiks minna ning kuna kõik komponendid on sama toote osad, siis töötavad need koos probleemideta. [14]
- Django aitab arendajatel tänu sisse ehitatud turvasüsteemidele vältida ka levinumaid turvaprobleeme, nagu näiteks *Clickjacking*, *Cross-Site Scripting* ja *SQL injection*. [13]

Negatiivsed aspektid [15], [16]:

- Django kasutades puudub vabadus ise midagi teha, on olemas Django arendajate nägemus, kuidas asjad peaksid toimima ning kui arendaja seda ei järgi, siis tõenäoliselt ei jõua rakenduse loomisega kaugemale. See tähendab, et arendajad peavad ka lihtsate programmide loomiseks paljud kontseptsioonid endale nullist selgeks tegema.
- Django on ebasobilik väiksemate projektide jaoks. See, mis oli positiivses pooles välja toodud, osutub ka negatiivseks, kuna kõik vajalik on raamistikuga kaasas, teeb see raamistiku väiksemate projektide jaoks liiga monoliitseks. Lisaks kasutab Django rakendus ka palju serveri ressursi.

4.2.3 Laravel

Laravel on PHP raamistik veebirakenduste arendamiseks, mida iseloomustab väljendusrikas ja elegantne süntaks. Laraveli arendajad usuvad, et arendus peaks olema nauditav ning loov. Sellepärast on üritatud lihtsustada tavapäraseid ülesandeid, millega tuleb kokku puutuda enamike veebirakenduse arenduse jooksul: näiteks autentimine, ruutimine ja sessioonid. Seda kõike on tehtud selliselt, et arendusprotsess oleks arendajale nauditav, kuid ilma rakenduse funktsionaalsust ohverdamata. [17]

Taylor Otwell, Laraveli looja, lõi Laraveli asenduseks CodeIgniter raamistikule, millel puudusid teatud võimalused, näiteks kasutajate autentimine ja autoriseerimine. [18]

Positiivsed aspektid:

- Põhjalik dokumentatsioon – loojad on kindlustanud, et nende kogu nende töö on arusaadav ja hästi kirjeldatud ning kaasas on selged juhised. [18]
- Tänu sellele, et Laravelil on ühilduvus PHP versiooniga 5.5 ja uuem, on võimalik kasutada kasutada ka uusimate PHP versioonide funktsioone. [18]
- Artisan – Laraveliga kaasasolev käsura tööriist, millega tuleb kaasa palju vajalikke käsked, millest on rakenduse loomisel abi. [19]

Negatiivsed aspektid [20]:

- Erinevate versioonide vahelise ühilduvuse puudumine.
- Raamistiku komponentide kvaliteet on väga kõikuv. On komponente, mis ei ole hästi disainitud.

4.2.4 Sobivaima raamistiku valik

Kuna raamistiku valimisega valib autor ka keele, milles tagarakendust arendama hakatakse, seejärel võrreldakse ka programmeerimiskeeli, milles raamistikud loodud on.

Valikus olevate raamistike loomiseks kasutatud programmeerimiskeelest on ainult Java staatiliselt tüübitud keel. Keel on staatiliselt tüübitud siis, kui muutuja tüüp on juba teada kompileerimise ajal, mitte programmi töötamise ja muutujale väärtuse omistamise hetkel. Tänu sellele on võimalik palju vigu tuvastada juba kompileerimisel. [21] Vähem vigu, mis töötavasse rakendusse jõuab, tähendab vähem potentsiaalseid arendaja tähelepanematuses tekkinud turvariske. Lisaks aitab see ka sisendi valideerimisel, kus vale tüüpi infot ei saagi muutujasse kirjutada.

RESTful API loomiseks on sobilikud kõik kolm raamistikku. Kuna bakalaureusetöö raames valmiv rakendus on pigem väikese suurusega, siis autor jätab kohe kõrvale Django, mis pole sobilik väiksemate projektide jaoks. [15], [16]

Järgmiseks võrdleb autor Spring Boot ja Laravel raamistikku (Tabel 1). Alltoodud tabelis suurem väärtus on parem.

Tabel 1. Spring Boot ja Laravel võrdlus

	Spring Boot	Laravel
Populaarsus	14.56	10.12
Autori kogemus	7	0
Kommuuni suurus	96.5k [22]	91,6k [23]

Populaarsuse puhul võrdles autor Stackoverflow poolt arendajate hulgas läbi viidud uuringus saadud statistikat. Kuna Spring Boot'i eraldi välja toodud ei olnud, siis kasutas autor võrdluses Spring raamistiku tulemust, sest Spring Boot on Spring raamistiku peale ehitatud, lihtsustamaks Spring raamistiku kasutamist. Väärtuseks on vastajate osakaal, kes andsid oma hääle vastavale raamistikule. [24]

Autor ei ole varasemalt Laraveliga kokku puutunud, seega kogemus antud raamistikuga puudub täielikult. Spring Bootiga on autoril varasemast kogemus olemas. Tabelisse läks väärtus, millega autor hindas enda varasemat kogemust antud raamistikuga kümne palli skaalas.

Kommuuni suurust hindas autor mõlema raamistiku Github'i repositooriumi statistikat vaadates. Autor liitis kokku projektidele antud tähed ning projektist loodud harud.

Tulemused on üsna sarnased, kuna pakutavate võimaluste poolest üsna sarnased raamistikud. Kuid sellegipoolest tundub eespool toodu põhjal, et kõige parem oleks kasutada Spring Boot raamistikku.

4.3 Eesrakenduse raamistiku valik

Eesrakendus on veebirakenduse osa, millega kasutaja otseselt suhtleb, kutsutakse ka kliendipoolseks rakenduseks. Sisaldab kõike, mida kasutaja otseselt lehte kasutades kogeb: teksti värve, pilte, graafikuid jne. [7]

Autor võrdleb kahte populaarset eesrakenduse loomiseks kasutatavat Javascripti raamistikku Angular ning React. Valik on tehtud KeM haldusala tehnoloogilise profiili põhjal. Angular ning React raamistike kompetents on KeMITis olemas. [6] Selle tulemusena jäävad välja teised populaarsed raamistikud, nagu näiteks Vue.js.

4.3.1 Angular

Angular on tasuta ning avatud lähtekoodiga raamistik veebirakenduste tegemiseks, mida arendab Google. Tegemist on komponentidel põhineva raamistikuga, millega saab luua erineva mastaapsusega rakendusi. [25]

Komponendi osadeks on TypeScripti klass, @Component() dekoraator, mis kirjeldab, kuidas komponenti mallis kasutada, HTML mall, mille järgi Angular teab, kuidas komponenti renderdada, ning valikuliselt CSS stiilid, mis kirjeldavad HTML elementide välimuse. [25]

Angulari plussideks on kahesuunaline andmete sidumine. Angulari rakenduste loomisel kasutatakse MVC mustrit, raamistik ise sünkroniseerib andmeid mudeli ja vaate vahel, tänu millele ei ole arendajal vaja selle jaoks eraldi koodi kirjutada. [26]

Sõltuvuste süstimine. Sõltuvused on teenused või objektid, mida klass töötamiseks vajab. Sõltuvuste süstimine tähendab seda, et vajaminevad objektid küsitakse välistest allikatest, mitte ei looda neid käsitsi. [27]

Kuna selle raamistiku taga on Google, mis pole väike tegija, siis tänu sellele on Angulari taha tekkinud ka suur arendajate kommuun. [28]

Miinusteks on järsk õppimiskõver ning ei sobi kõige paremini algajate jaoks. Tegemist on arvamuslimelise raamistikuga. See tähendab, et raamistiku kasutamisel on omad reeglid ja põhimõtted, mis tuleb arendajal endale selgeks teha ja neid järgida. [29]

Angular on ka liiga üksikasjalik (inglise keeles *verbose*) ja keeruline. Angularis kasutatakse komponente, kuid komponentide haldamine on tehtud väga keeruliseks. Tulemusena tuleb Angularis arendades teha palju korduvaid tegevusi. [26]

4.3.2 React.js

React.js on loodud suurt jõudlust silmas pidades. Teeki arendab Meta, varasemalt tuntud Facebook-i nime all. Jõudlus tuleb virtuaalse DOM-i kasutamisest. Virtuaalne DOM on koopia originaalsest DOM-ist, mida hoitakse mälus. Kui tehakse muudatus virtuaalses DOM-is, võrreldakse seda viimase hetktõmmisega, sellega tehakse kindlaks, mis komponendid muutusid ja kui on teada, mida vaja muuta on, siis tehakse muudatus ka originaalses DOM-is. Kuna DOM manipuleerimine on ressursinõudlik operatsioon, tehakse nii DOM-is muudatusi võimalikult vähe. [30]

React.js rakenduste loomisel kasutatakse CBA mustrit. See tähendab seda, et kasutajaliidese erinevad osad (komponendid) on kapseldatud iseseisvateks osadeks, mis kõik haldavad oma olekut. Komponendid on taaskasutatavad ning neid saab vastavalt vajadusele lisada/eemaldada. [31]

React.js puhul ei ole tegemist raamistikuga, vaid teegiga. Seda sellel põhjusel, et kuna raamistikus on olemas kõik ühe tervikliku rakenduse tegemiseks, siis React.js puhul saadakse võimekad Reacti ja React DOM teegid, mille abil saab JSX süntaksit kasutades kasutajaliidest luua. Sellele lisaks veel olekuhalduse vahendid. Sellega aga asi piirdub ning paljud teegid, mis näiteks Angularil on juba baaspaketis sees, Reactis puuduvad ja tuleb eraldi lisada. [32] Vaatamata sellele on React.js Stackoverflow's iga-aastase

arendajate seas tehtava küsitluse andmetel 2021. aasta kõige populaarsem veebirakenduste loomise platvorm. [24]

4.3.3 Sobivaima raamistiku valik

Mõlemale platvormile on võimalik bakalaureusetöö käigus valmivat eesrakendust kirjutada. Angular on täiemõõtmeline raamistik, mis tähendab, et enamus rakenduse loomiseks vajalikust funktsionaalsusest on juba raamistikku sisse ehitatud. Reacti puhul tuleb kasutada lisaks veel erinevaid teeke, nagu näiteks React Router.

Autor võrdleb omavahel Angulari ning Reacti (Tabel 2), et rakenduse loomiseks sobivam välja selgitada. Tabelis suurem väärtus on parem.

Tabel 2. Angular ja React võrdlus

	Angular	React
Populaarsus	22.96 [24]	40.14 [24]
Õppimiskõver	Järsk [33]	Mõõdukas [33]
Autori kogemus	2	6
DOM	Reaalne [33]	Virtuaalne [33]
Kommuuni suurus	101,8k [34]	224,1k [35]

Populaarsuse määramisel kasutas autor StackOverflows tehtud arendajate seas läbi viidud küsitlust ning võttis väärtuseks küsitluses raamistiku poolt saadud vastajate häälte osakaalu.

Õppimiskõver ja autori kogemus on olulised sellepärast, et autor pole eesrakenduse raamistikega väga kogenud, see tähendab, et tuleb jooksvalt ka õppida. Tulebki varakult välja selgitada, kui palju aega õppimisele kuluda võib. Oma kogemust hindas autor kümnepalli skaalas. Autori kogemus Angulariga on väike ning raamistiku õppimiskõver on järsk võrreldes Reacti keskmise kogemusega ning mõõduka õppimiskõveraga. Autoril tuleks Angulari hakata õppima üsna algtasemelt, samas Reacti puhul on põhitõed selged.

Tänu DOM virtualiseerimisele on andmete uuendamine React rakenduses kiirem. [33] Kuigi Angularis DOM pole virtualiseeritud, on seal kasutusel muudatuste tuvastamine,

mis tuvastab sündmuseid brauseris, mille põhjal otsustatakse, kas komponendi vaadet on vaja uuendada või mitte. [36]

Kommuuni suuruse puhul võttis autor kasutusele Githubi repositooriumis projektile jäetud tähtede arvu ning projektist loodud harude arvu summa.

Tabeli järgi on sobivam eesrakenduse loomiseks React.js.

4.4 REST arhitektuuriline stiil

Termini REST (*REpresentational State Transfer*) võttis kasutusele Roy Fielding aastal 2000. See on arhitektuuriline stiil, rakenduste üle võrgu ühendamiseks ning on kasutusel veebiteenuste arendusel. REST ei sunni peale ühtegi reeglit kuidas midagi teha madalamal tasemel. See annab kõrgema taseme juhtnöörid, konkreetse lahenduse valimine on arendaja valida. [37]

REST defineerib 6 arhitektuurilist piirangut, mis teevad veebiteenusest RESTful API [37]:

- Ühtne liides - rakendades üldsuse põhimõtet liidestele, lihtsustatakse üleüldist süsteemi arhitektuuri.
- Klient-server - muster sunnib peale vastutusala eraldamise, mis aitab kliendi ja serveri komponente üksteisest sõltumatult arendada.
- Olekuta – kõik kliendi päringud serverile peavad sisaldama kogu vajalikku informatsiooni konkreetse päringu salvestamiseks.
- Vahemällu salvestatav – kliendil on õigus andmeid sama või samaväärsete päringute puhul määratud aja jooksul uuesti kasutada.
- Kihiline süsteem – kihilises süsteemis iga komponent näeb ainult seda kihti, millega ta vahetult suhtleb;
- Kood nõudmisel (valikuline) – REST võimaldab kliendi funktsionaalsust laiendada ka koodi allalaadimise ja käivitamise teel. See võimaldab osa funktsionaalsusest hoida serveri poolel.

Andmeid käsitletakse REST arhitektuurilises stiilis ressurssidena. REST arhitektuur ei määra seda, mis vormis ressursse edastatakse. See on arendaja valida. Samamoodi ei määrata seda, mis protokollid edastamisel kasutatakse. [37]

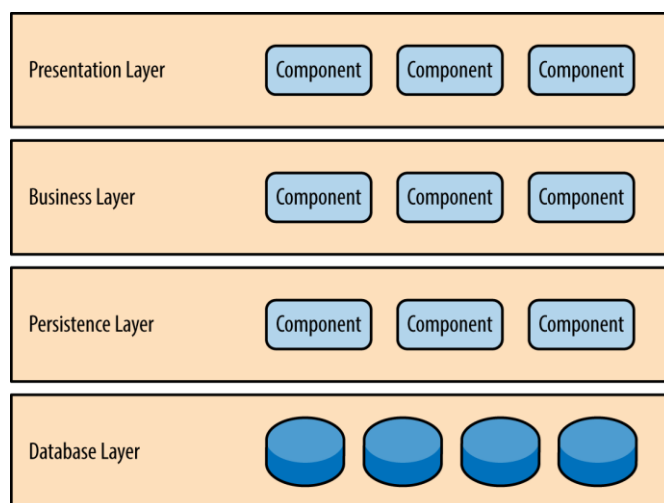
4.5 Tarkvaraarhitektuuri mustrid

Käesolevas peatükis vaatleb autor lähemalt kahte tarkvara arendamisel kasutatavat mustrit ning valib sobivaima mustri, mida kasutada tagarakenduse arendamisel.

4.5.1 Kihiline arhitektuur

Kihiline arhitektuur on meetod koodi jagamiseks kihtideks, millest igal on süsteemis kindel roll. Kihtide arvu pole otseselt määratud, kuid tavaliselt kasutatakse nelja kihti (joonis 1) [38]:

- Presentatsioonikiht, mis tegeleb päringute vahendamisega - võtab vastu päringuid ning tagastab vastuse.
- Äriloojika kihi ülesandeks on äriloojika põhjal andmete töötlemisega.
- Andmete püsivuse kiht suhtleb andmebaasiga.
- Andmebaas talletab andmeid.



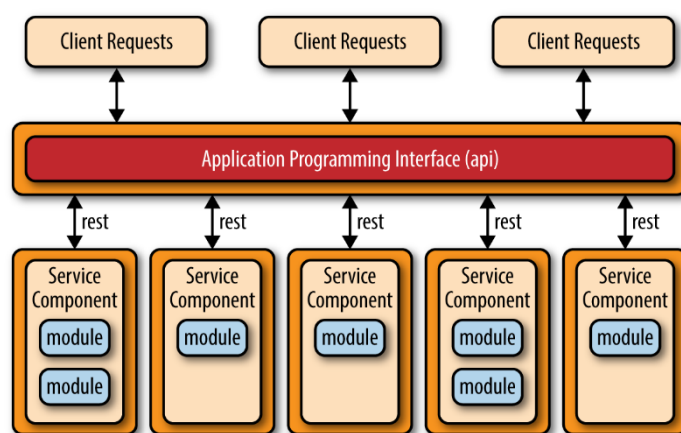
Joonis 1. Kihilise arhitektuuri skeem [38]

Iga kiht suhtleb ainult endast allapoole jääva kihiga. Antud mustri tugevuseks on see, et iga kiht tegeleb ainult sellega loogikaga, mis jääb tema vastutusalasse. Näiteks äriloojika

kihis asuvad komponendid ei tegele andmebaasi suhtlusega. See tagab meile selle, et kui me muudatusi teeme, jäävad need konkreetse kihi piiresse ja meil ei ole näiteks andmete püsivuse kihis tehtud muudatuse tulemusena vaja hakata parandama presentatsioonikihti. [38]

4.5.2 Mikroteenuste arhitektuuri muster

Rakendus koosneb eraldiseisvatest ning eraldi evitatavatest komponentidest. Sisuliselt on tegemist mitmete väiksemate üksteisest sõltumatute rakendustega, millega suhtleb API kiht. (joonis 2) [38]



Joonis 2. Mikroteenuste arhitektuuri skeem [38]

Antud mustri tugevuseks on pakutav modulaarsus ning komponentide sõltumatus üksteisest. Monoliitses rakenduses on rakenduse komponendid kõik ühe evitatava üksuse osa, mis teeb süsteemi üsna paindumatuks ning iga kord, kui vaja rakendust evitada, on kogu rakendus maas. Mikroteenuste arhitektuuri muster lahendab selle probleemi nii, et kõik komponendid on iseseisvad, neid arendatakse, testitakse ning evitatakse eraldi. Kui ühe komponendiga on probleem, siis ei mõjuta see teiste komponentide tööd. [38]

4.5.3 Sobivaima mustri valik

Välja toodud kaks mustrit pole ainsad, kuid autori meelest olid need kaks sobivaimad bakalaureusetöös käigus valmiva tagarakenduse loomiseks. Tõenäoliselt oleks loodav rakendus oma väiksuse tõttu sobiv mikroteenuste arhitektuuri mustri juures üheks komponendiks. Ka evituse protsess on ühe monoliitse rakenduse puhul lihtsam. Sellepärast on mõistlikum kasutada kihilist arhitektuuri.

4.6 Rakenduste turvalisuse kontrollimise standard

OWASP ASVS (*Open Web Application Security Project Application Security Verification Standard*) projekt loob aluse veebirakenduste tehniliste turvakontrollide tegemiseks ning pakub arendajatele ka loendit turvalise arenduse nõuetest. [39]

Projektiga üritatakse normaliseerida turvatestimise turgu, muutes kättesaadavaks avatud turvalisuse standardi, mida võib kasutada tõstmaks veebirakenduste usaldusväärset. Nõuded arendati välja järgmisi eesmärke silmas pidades [39]:

- Standardiga saab anda konkreetseid mõõdikuid, millega saab turvalisust mõõta.
- Suunaja - anda arendajatele juhiseid turvameetmete kohta, mida peaks lisama, et rakendus vastaks turbenõuetele.
- Hangetes kasutatav – et oleks alus, millele viidata, kui hangetes on vaja turvanõudeid kirjeldada.

4.6.1 Kuidas ASVS rakendada

Kõige parem on standardit rakendada, käsitledes seda kui skeemi, mille põhjal oma rakenduse vajadustele põhinevad turvanõuded paika panna.

ASVS defineerib ise kolm rakenduste turvataset [40]:

- Taseme 1 saavutab rakendus siis, kui ta on kaitstud kõikide lihtsalt avastatavate turvanõrkuste vastu ning nende riskide vastu, mis on kirjeldatud näiteks OWASP 10 populaarsema riski loendis. Kokku on esimese taseme kohta loetletud 127 nõuet. See tase peaks olema miinimum kõigi rakenduste jaoks. Tase 1 on testitav automaatselt mõne tööriistaga ning ilma lähtekoodile ligipääsu omamata.
- Taseme 2 saavutab rakendus siis, kui ta on kaitstud enamuse riskide vastu, mis tänapäeva tarkvara ohustavad. Mõeldud tarkvarale, mis tegelevad oluliste ärikriitiliste funktsioonidega, töötlevad tervishoiuga seonduvat informatsiooni või on seotud mõne muu sensitiivset infot omava valdkonnaga.
- Tase 3 on reserveeritud rakendustele, milles on turvalisus eriti oluline, nagu näiteks militaarvaldkonnas, tervishoius, kriitilise infrastruktuuri puhul jne.

Rakendused, kus teostatakse kriitilisi funktsioone, mille töökindlus ja selle puudumine võib mõjutada oluliselt organisatsiooni operatsioone või isegi elusid.

Bakalaureusetöö käigus loodav rakendus peaks vastama taseme 1 nõuetele, kuna rakenduses ei käidelda sensitiivset infot ning tegemist pole ärikriitiliste funktsioonidega.

ASVS nõuded on jagatud vastavalt valdkondadele peatükkideks, mis omakorda on jagatud osadeks. Igas osas olevad nõuded on järjest nummerdatud. Kui on vaja mõnele konkreetsele nõudele viidata, siis saab seda teha kujul <peatükk>.<osa>.<nõue>, näiteks 1.11.3. Kuna nõuded võivad muutuda ka standardi erinevate versioonide vahel, siis võib kõige ette lisada ka ASVS versiooni kujul v<ASVS versioon>-<peatükk>.<osa>.<nõue>, näiteks v4.0.3-1.11.3. Versiooninumbri ees olev „v“ on väiketäht. [40]

ASVS peatükid:

1. Arhitektuur, disain ja ohtude modelleerimine;
2. Autentimine;
3. Sessioonihaldus;
4. Juurdepääsukontroll;
5. Valideerimine, puhastamine ning kodeerimine;
6. Talletatud krüptograafia;
7. Vigade käsitlemine ning logimine;
8. Andmekaitse;
9. Kommunikatsioon;
10. Pahatahtlik kood;
11. Äriloogika;
12. Failid ja ressursid;
13. API ja veebiteenus;

14. Konfiguratsioon.

Bakalaureusetöö käigus loodava rakenduse puhul võib mitme peatüki nõuded kõrvale jätta. Rakenduses ei ole erinevaid kasutajaid ning rolle, seega ei ole vaja mõelda autentimisele, kasutajate autoriseerimisele ning parooli turvalisusele. Samuti ei kasutata loodavas rakenduses sessioone.

Kuna loodavas rakenduses on oluline roll kasutajate sisestatud infol nii otsinguvormidesse sisestatud teksti kui ka kalkulaatoritesse sisestatud andmete näol, siis tuleb tähelepanelikult jälgida ASVS peatükis 5 antud soovitusi, kuidas kasutajate sisendid puhastada ning valideerida. Lisaks on rakenduse kasutajatel võimalus infot failist üles laadida ning pilte alla laadida. Selle puhul tuleb jälgida, et rakendus ei võtaks väga suuri faile vastu, mis võimaldaks serveriketta täis kirjutada. Veel tuleks jälgida failinime, et seda ära kasutades ei oleks võimalik ligi pääseda muidu keelatud ressurssidele. Näiteks võib failinimi sisaldada käskke, mis käivituvad, kui fail mõnele tööriistale argumendina ette anda. Kuna loodav rakendus kasutab osade kalkulaatorite puhul väliseid tööriistu, mille sisend läheb failist, siis see on kindlasti aktuaalne. [41]

Tähtis on ka logimine. Logide põhjal on võimalik muu hulgas tuvastada süsteemi vastu tehtud rünnakukatseid. ASVS käsitleb logimist peatükis 7.

4.7 Veebiturbe testimise juhend

WSTG (*Web Security Testing Guide*) on juhend veebirakenduste ja -teenuste testimiseks. Projekti eesmärgiks on aidata inimestel aru saada, „mida“, „miks“, „kuna“, „kus“ ja „kuidas“ veebirakendusi testida. Tulemuseks on täielik raamistik, milles kirjeldatakse raamistikku üldiselt, ning tehnikaid, kuidas raamistikku praktikas rakendada. Raamistikku valideerivad pidevalt praktiseerivad turvaekspertid. [42]

WTSG peatükk 3 kirjeldab OWASP testimise raamistikku ning seletab tehnikaid ja ülesandeid sõltuvalt tarkvaraarenduse elutsükli etapist. Peatükk 4 käsitleb konkreetsete haavatavuste (näiteks *SQL injection*) testimist koodi kontrollimise ja läbistustestimise teel. [42]

WSTG peatükk 4 iga alampeatükk sisaldab infot konkreetse nõrkuse kohta, kus antakse kõigepealt ülevaade selle olemusest. Kirjeldatakse testi eesmärke ning seejärel tuleb detailne juhend, kuidas testida. Osades peatükkides on info ka vastumeetmete kohta.

4.8 Kaardimooduli teegi valik

Käesolevas peatükis leitakse sobivaim teek kaardimooduli loomiseks.

4.8.1 Leaflet

Leaflet on vabavaraline JavaScripti teek, mida kasutatakse interaktiivsete kaardirakenduste loomiseks. See on loodud kasutatavust, lihtsust ja jõudlust silmas pidades ning töötab kõikide suuremate töölaua ja mobiilsete platvormide peal. [43]

Leaflet toetab vaikesi ainult kolme projektsiooni: EPSG:4326, EPSG:3857 ja EPSG:3395, kuid ei toeta Eestis kasutatavat EPSG:3301 projektsiooni. [44]

Kaardiprojektsioon on viis kujutada sfäärilist pinda tasapinnal. Kaardiprojektsiooni võib ka nimetada matemaatiliseks funktsiooniks geograafiliste koordinaatide ja tasapinnaliste rist- või ka polaarkoordinaatide vahel. Iga projektsiooni kirjeldatakse temale iseloomulike valemitega. [45]

EPSG:4326 on kõige lihtsam projektsioon, kus x on pikkuskraad ja y laiuskraad. Seda projektsiooni kasutavad GPS süsteemid. [46]

EPSG:3857 ehk Mercatori projektsioon on levinud silindiline projektsioon, mida kasutavad enamus tasuta või tasuliste kaardiandmete levitajad. Kasutusele võttis Gerardus Mercator juba aastal 1569. Kaardil on pikkuskraade kujutatud üksteisega paralleelselt, mis tähendab, et kaardil on moonutused - mida kaugemal ekvaatorist, seda suuremad. [44], [47]

EPSG:3395 eeldab, et Maa on ellipsoid, on komplekssem kui EPSG:3857. Kasutatakse Euroopa-keskset vaadet, väiksema skoobiga kaardistuse tegemiseks. [44], [48]

Leaflet on disainitud selliselt, et soovitud tulemuseni jõutakse minimaalse hulga koodiga. Seda soovitatakse lihtsamate kaardirakenduste tegemiseks. [49]

4.8.2 OpenLayers

Javascripti teek, mis on loodud veebilehtedele dünaamiliste kaartide lisamiseks, võimaldab näidata mistahes allikast pärinevaid kaardiruute, vektorkujul andmeid ning markereid. [50]

Mõeldud on pigem keerukamate GIS rakenduste arenduseks, tänu millele on ka kommuun väiksem kui Leaflet'il. [49] Nagu Leaflet, ei toeta ka OpenLayers vaikimisi EPSG:3301 projektsiooni. Ilma defineerimata toetatakse vaid EPSG:4326 ja EPSG:3857 projektsioone.

Kõik vajalik on teegis olemas vajaduseta lisaks pistikprogramme paigaldada.

4.8.3 X-GIS2 kaardirakendus

Maa-ametil on olemas X-GIS 2 kasutajaliides, mille kaudu saab ligi erinevate andmekogude ruumi- ja kaardistusandmetele, sealhulgas ka geodeetiliste punktide andmekogu kaardikihtidele. [51]

4.8.4 Kaardikihid

Kuna Maa-amet pakub ka ise WMS teenust, siis autor ei vaata väliste teenusepakkujate poole nagu näiteks OpenStreetMap. Maa-ameti WMS teenuse pakutavate kihtide hulgas on ka kõik geodeetiliste punktide andmekogu erinevat tüüpi punkte kuvavad kihid:

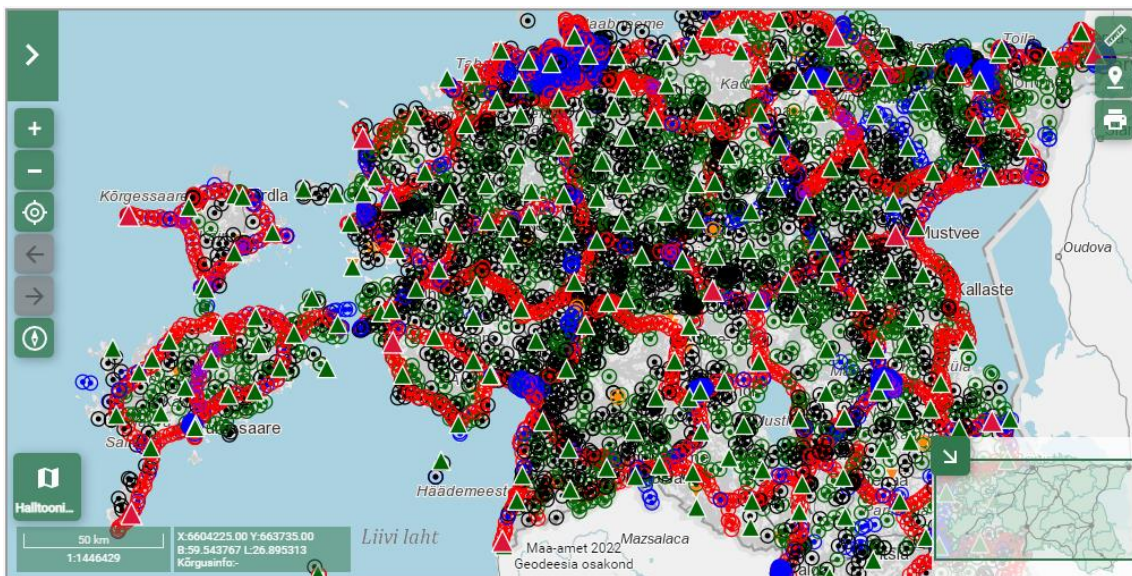
- Riiklik geodeetiline võrk;
- Gravimeetriline võrk;
- Kohalik geodeetiline võrk;
- Kõrgusvõrk;
- Geodeetiliste märkide seisukorra teemakaart.

Aluskaardiks kasutab autor halltoonides kaarti. Maa-ameti pakutavate WMS ja WFS teenuste loetelu on ära toodud Maa-ameti Geoportaalil. [52] Samas kohas on võimalik ka näha iga teenuse GetCapabilities päring, mis näitab teenuse kohta metainfot, nagu näiteks toetatud operatsioonid ning parameetrid ja teenuse pakutavad kaardikihid. [53] Teenused

on tasuta kasutatavad ning Maa-ameti poolt on valminud ka kasutusjuhendid WMS teenuse erinevates tarkvarades kasutamiseks. [54]

4.8.5 Sobivaima teegi valik

X-GIS2 kaardirakenduse saab lisada lehele kasutades HTML IFrame elementi. Lähemal katsetamisel veebilehitseja ajaloos edasi-tagasi liikumisel tekkis probleem, kus rakendus ei suutnud meelde jätta kuvatava piirkonna suurust ja kuvas kogu Eesti kaardi. Kuna punkte on palju, siis ei ole midagi antud kaardilt aru saada (joonis 3).



Joonis 3. X-GIS2 probleemi näide

Kuna X-GIS2 kaardirakenduses on lisaks punktide kuvamisele ka palju lisafunktsionaalsust, siis arvas autor, et on mõistlikum punktide kuvamiseks luua OpenLayers või Leaflet teeki kasutades eraldi kaardimoodul ainult punkti asukoha kaardil näitamiseks.

Kuna kumbki raamistik ei toeta vaikesi EPSG:3301 projektsiooni, tuleb see käsitsi defineerida. OpenLayers teegis saab seda teha järgnevalt (joonis 4).

```
proj4.defs(
  'EPSG:3301',
  '+proj=lcc +lat_1=59.3333333333334 +lat_2=58 +lat_0=57.51755393055556
+lon_0=24 +x_0=500000 +y_0=6375000 +ellps=GRS80 +towgs84=0,0,0,0,0,0
+units=m +no_defs'
);
```

Joonis 4. OpenLayers teegis EPSG:3301 projektsiooni defineerimine. [55]

Leafletis toimib projektsiooni defineerimine järgnevalt (joonis 5):

```
const projection = new L.Proj.CRS(
  'EPSG:3301', '+proj=lcc +lat_1=59.3333333333334 +lat_2=58
+lat_0=57.51755393055556 +lon_0=24 +x_0=500000 +y_0=6375000
+ellps=GRS80 +towgs84=0,0,0,0,0,0 +units=m +no_defs',
  {resolutions: [4000, 2000, 1000, 500, 250, 125, 62.5,
31.25, 15.625, 7.8125, 3.90625, 1.953125, 0.9765625,
0.48828125, 0.244140625, 0.122070313, 0.061035156,
0.030517578, 0.015258789]},
  origin: [40500, 5993000],
  bounds: L.bounds([40500, 5993000], [1064500, 7017000])
});
```

Joonis 5. Leaflet teegis EPSG:3301 projektsiooni defineerimine. [55]

Seega pole EPSG:3301 projektsiooni toetuse puudumine kummagi teegi puhul takistuseks.

Leaflet on populaarsem. Google otsing „leaflet stackoverflow“ andis 872 000 vastet, võrdluseks „openlayers stackoverflow“ andis 123 000 vastet.

Kuna nii Leaflet kui ka OpenLayers teegid mõlemad olid sobivad, sai valikutegemisel otsustavaks Leaflet'i lihtsus ja populaarsus. Täpsemalt react-leaflet teegi lihtsus, mille React'i komponente kasutades oli väga lihtne erinevate kihtidega kaart valmis teha. Olemas on ka sarnane teek OpenLayers'i jaoks – react-openlayers, kuid kuna selle projekti juures viimane tegevus oli 2 aastat tagasi, siis seda ei olnud otstarbekas kasutada. [56]

4.9 Kalkulaatorid

Osad kalkulaatorid on olemas exe failidena. Nende rakenduste puhul on C keeles kirjutatud kood ka olemas, töö autoril aga C keelega kogemus puudub ning ilma konteksti tundmata raske aru saada. Sellepärast ei hakanud autor seda koodi Javasse ümber

kirjutama, vaid võttis plaani läbi loodava rakenduse nende tööriistade käivitamise ning väljundi töötlemise Java objektiks.

Algselt oli autoril plaanis need välise tööriistana olemasolevad kalkulaatorid panna Spring Boot rakenduse klassi ressursside alla. Lähemal uurimisel tuli välja, et sealt ei saa väliseid tööriistu käivitada, kui rakendus on jar failist käima pandud. Jar faili puhul on tegemist platvormist sõltumatu failiga, mis sisaldab java klasse ja kogu nendega seotud metainfot ning ressursse. Kui jar failist exe faili käivitada, siis seal asuvat exe faili ei leia operatsioonisüsteem üles. [57], [58]

Üheks variandiks oli vastavalt vajadusele need failid enne käivitamist kuskile ajutisse kausta kopeerida ning sealt käima panna ja pärast kustutada, kuid see ei tundunud hea ideena. Kui päringuid peaks korraga palju tulema, tekitab selline kopeerimine/kustutamine serverile lisakoormust. Autor otsustas jätta need tööriistad jar failiga samas kaustas olevasse alamkausta, kust leiab alati relatiivselt jar faili asukohaga need tööriistad üles.

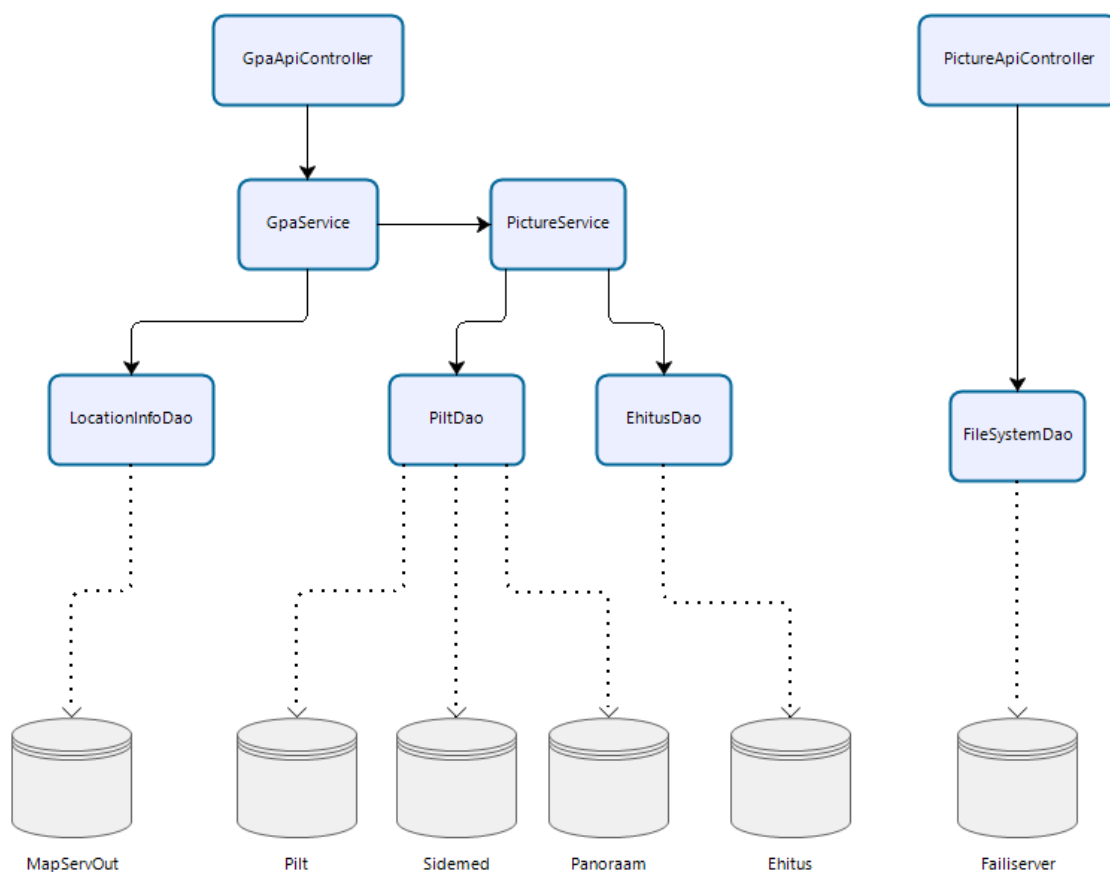
Nende kalkulaatorite jaoks, mis ei kasutanud eraldiseisvaid tööriistu, kaalus autor tööle panemist eesrakenduses. Testimiseks valmis ka üks kalkulaator, mille autor tegi TypeScriptis ning see töötas ainult eesrakenduses. Arvutusi tehes tekkisid aga ebatäpsused. Kuni neli-viis kohta peale koma oli tulemus sama mis olemasoleva GPA rakendusega arvutades, kuid nendest kohtadest edasi läks tulemus valeks. Autor kasutas ka EPSG.io lehte kontrollimiseks, millega võrreldes oli tulemus sama. Javas seda probleemi ei olnud. Lisaks oli vaja kalkulaatorid ka läbi API kättesaadavaks teha, seega polnud mõtet kahte kohta sisuliselt sama funktsionaalsust luua. Nii jäid kalkulaatorid ainult tagarakendusse, mida kasutab ka eesrakendus.

5 Lahenduse teostus

Käesolev peatükk kirjeldab rakenduse loomise praktilist osa.

5.1 Tagarakendus

Autor lõi bakalaureusetöö käigus tagarakenduseks Spring Boot rakenduse, mis serveeris API lõpp-punkte (joonis 6). Rakendus sai loodud kihilise arhitektuuri mustrit kasutades. Arvutusteks kasutatavad algoritmid sai autor analüüsidest varasemat rakendust ning pani need tööle Javas.



Joonis 6. GPA tagarakenduse skeem, ilma kalkulaatorite API lõpp-punktideta

GpaApiController ülesandeks on võtta vastu päringud klientrakendusest.

GpaService leiab läbi LocationInfoDao info geodeetiliste punktide kohta ning seejärel otsitakse piltide ID-d. Vastavalt ID-le kontrolliti, kas pilt reaalselt ka failiserveris eksisteeris. Kui leiti failiserverist ID-le vastav pilt, siis lisati info vastusesse, mis kontrolleri kaudu kliendile vastuseks läks.

PictureApiControlleri ülesandeks oli serveerida pilte. GpaApiControllerist läks vastuseks info ka piltide kohta. Kui klient soovis pilti näha, siis tehti päring PictureApiControllerisse, mis siis luges failiserverist sisse pildifaili (joonis 7). Kui tegemist oli PDF failiga, siis PictureService teisendas PDF faili JPG failiks ning saatis kliendile vastusena.

```
@GetMapping(value = "/image/{type}/{id}", produces = MediaType.IMAGE_JPEG_VALUE)
public @ResponseBody byte[] FindImages(@PathVariable Long id, @PathVariable String type) {
    return service.loadImage(id, type);
}
```

Joonis 7. Koodinäide API kaudu pildi saatmisest

Pildi saatmisel oleks olnud veel variandiks luua JSON objekt ning pildifaili sisu base64'ga kodeerida ning saata JSON-i lastina. See oleks olnud sobiv variant, kui pildiga koos oleks vaja olnud ka näiteks metaandmeid saata või kohe läbi GpaApiControlleriga tehtud päringu vastusesse piltide ID-de asemel pildid kaasa panna. Kuid kuna metaandmeid ei olnud ning pilte alati ei vaadatagi, siis oli otstarbekas saata pilt eraldi, kui kasutaja pilti vaadata soovis. Lisaks tuleks base64'ga kodeeritud pilt umbes 25% suurem. [59]

Failide PDF-ist JPG vormingusse teisendamisel kasutas autor vabavaralist Apache PDFBox teeki.

Piltide suuruste korrigeerimiseks kasutas autor Imgscalr teeki, mis kasutab oma töö tegemisel AWT teeki kuuluvat Graphics2D klassi. Seda kasutades sai lihtsalt valida, kas eesmärgiks on pildi suuruse muutmise kiirus või tulemuse kvaliteet. [60] Kasutades parameetrit *Scalr.Method.QUALITY*, oli tulemus hea (joonis 8).

PANORAAM

ID nr: 53-883-2810

Punkti nr: 2810

Tsentri nr: 37032

Punkti nimi: Sassi

Märgi tüüp: 5001

Välisvormistus: Tunnuspost, kupits

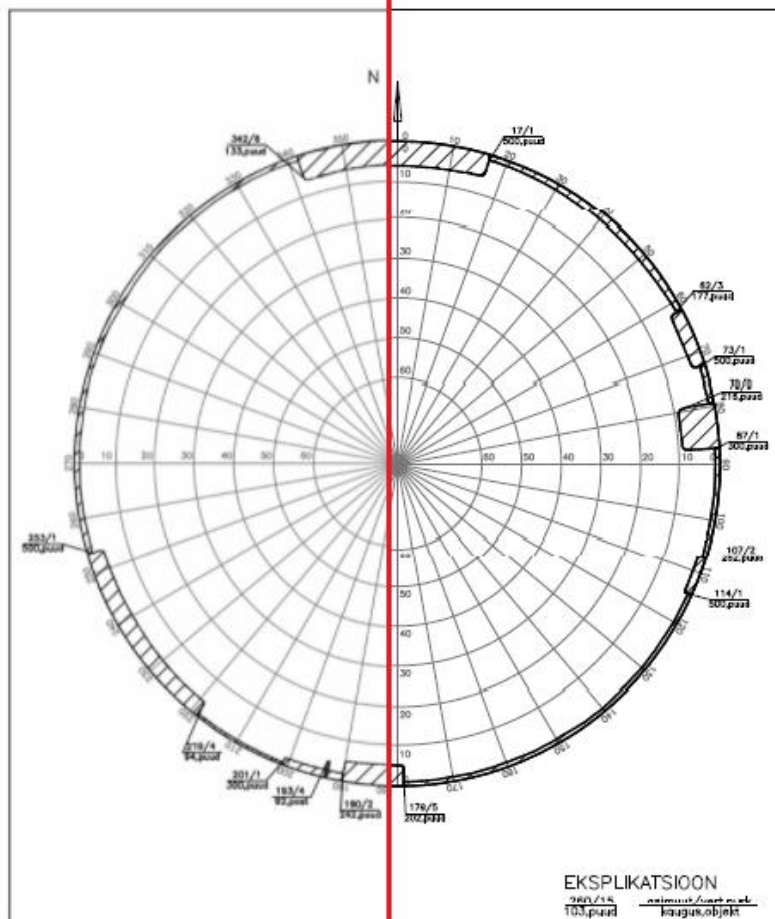
Asukoha kirjeldus: Sassi külast 1.2 km lõunasse, Ange talust 0,6 km loodesse, põllul.

Järk: Tihendusvõrk

Tsenter maapinnast: -

Koordinaadid: X= 6489280,975

Y= 563364,143



EKSPLIKATSIOON

280/1/4 - valmis/kuus m.a.k.

103.puud - kaugus,objekt

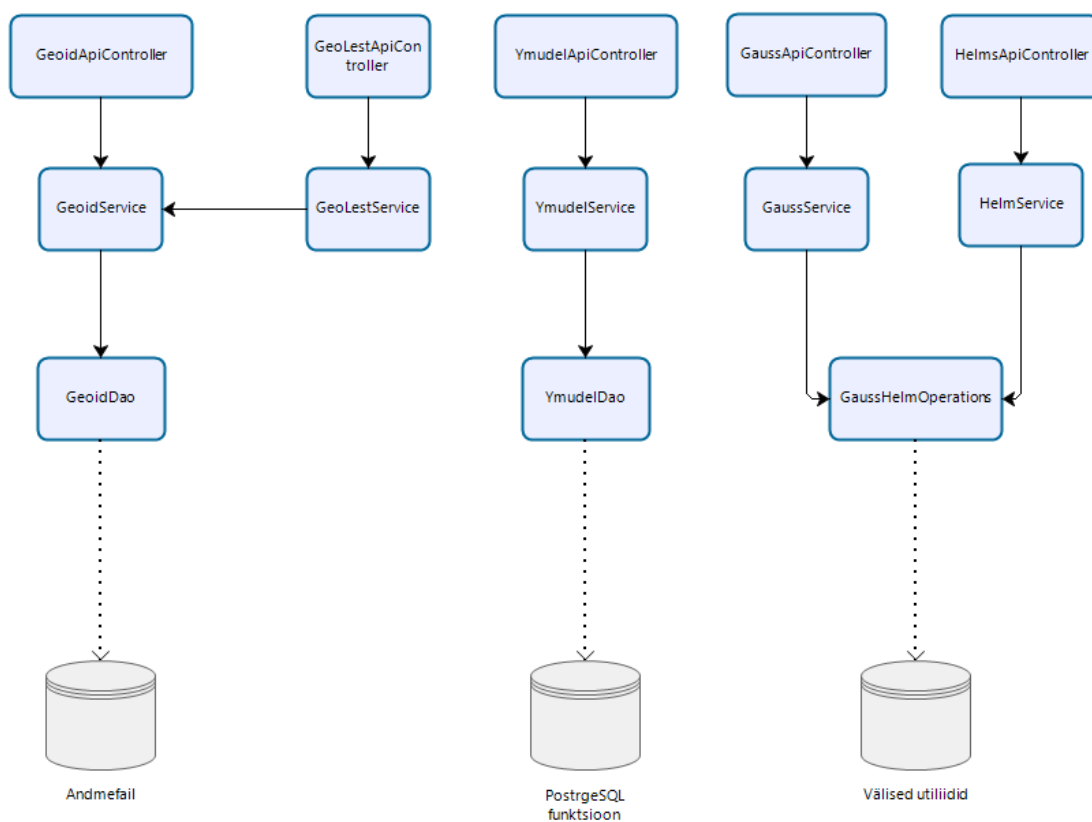
Koostas: Raili Sõna

Koostatud: 09.2020

Geo S,T, OÜ

Joonis 8. Kvaliteedi võrdlus. Vasakul teisendatud JPG, paremal originaal PDF

Kalkulaatorite jaoks loodi viis API lõpp-punkti (joonis 9).



Joonis 9. Kalkulaatorite API skeem

- GeoidApiController, võtab vastu geoidi arvutamiseks algandmed ning peale arvutamist tagastab vastuse. Arvutamisel kasutatakse geodeetiliste koordinaatidega geoidimudelit. Geoidi mudeli EST-GEOID2017, andmed saadakse binaarfailist. Mudel on välja arvutatud Maa-ameti poolt koostöös Tallinna Tehnikaülikooliga aastal 2017. [1], [61] Vastavalt sisendkoordinaatidele arvutatakse asukoht failis, millest alates tuleb binaarfailist sisse lugeda neli baiti

ehk 32 bitti, need teisendatakse ujukomaarvuks (joonis 10).

```
int z3Bits = (gridFile[pos + 7] & 0xff) << 24 |
             (gridFile[pos + 6] & 0xff) << 16 |
             (gridFile[pos + 5] & 0xff) << 8 |
             (gridFile[pos + 4] & 0xff);
Float z3 = Float.intBitsToFloat(z3Bits);
```

Joonis 10. Sisseloetud baidijadast baitide lugemine ja 32-bitise ujukomaarvu loomine

- GeoLestApiController võtab vastu sisendkoordinaadid, mida tuleb teisendada, ning GeoLestService teenus arvutab tulemuse ja kontrolleri saab vastuse tagasi.
- YmudelApiController võtab vastu sisendkoordinaadid ja tagastab vastuse. YmudelService teenus korrastab andmeid ning YmudelDao kasutab PostgreSQL andmebaasis olevat funktsiooni tulemuse arvutamiseks.
- GaussApiController ja HelmApiController võtavad samuti vastu sisendandmed ning tagastavad vastuse. GaussService ning HelmService valmistavad andmed ette ning genereerivad käivituseks vajalikud käsurea argumendid sisendi põhjal (joonis 11). GaussHelmOperations loob sisendfailid välise utiliitide jaoks. Seejärel käivitatakse utiliidid. GaussService ja HelmService töötlevad utiliitide väljundi, mis on lihtsas sõne vormingus, objektideks.

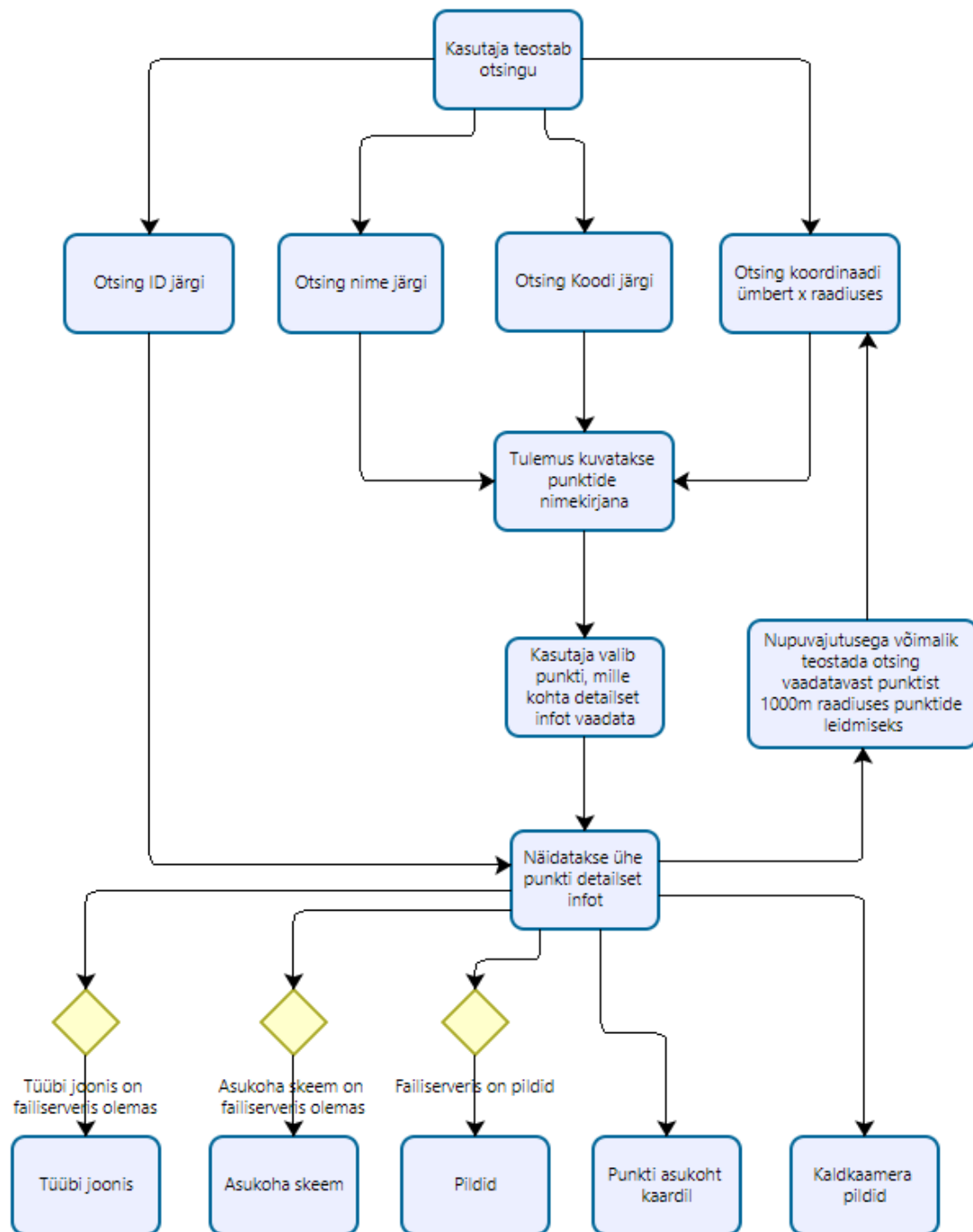
```
String[] commands = new String[]{gaussHelmOperations.getAppPath() + "\\tools\\gaustr\\gauss97.exe",
    gaussHelmOperations.getAppPath() + "\\tools\\gaustr\\" + fileNameMapper.getFileName(input.getSourceFormat()),
    gaussHelmOperations.getAppPath() + "\\tools\\gaustr\\" + fileNameMapper.getFileName(input.getResultFormat()),
    inputFile.getAbsolutePath(),
    outputFile.getAbsolutePath()
};
```

Joonis 11. Utiliidi kaustateed ning käsurea argumendid

5.2 Eesrakendus

Eesrakenduseks loodi bakalaureusetöö käigus üheleherakendus React.js'is. Üheleherakenduse puhul muudetakse dünaamiliselt ühte lehte, selmet serverist terveid lehti alla laadida.

Esimeseks tegevuseks oli rakenduse funktsionaalsuse kaardistamine. Selleks sai loodud voodiagramm (joonis 12).

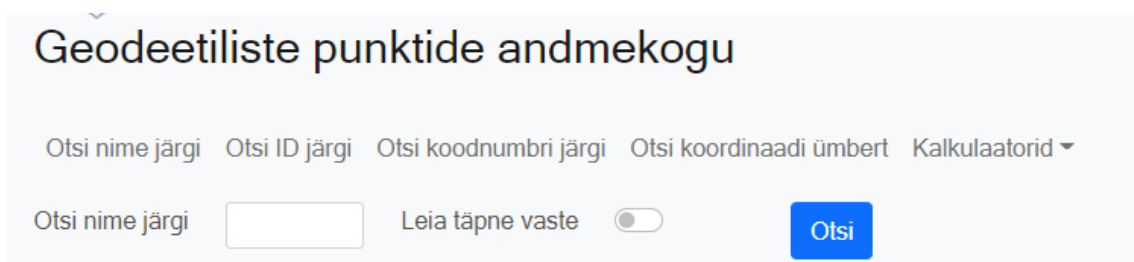


Joonis 12. Eesrakenduse voodiagramm

Eesrakenduse loomisel kasutati kasutajaliidese loomisel React Bootstrap teeki, mis teeb Bootstrap CSS raamistiku võimekuse React.js komponentide näol React.js projektis kasutatavaks. [62] Autor vaatas ka alternatiivseid React.js komponentidena kasutatavaid kasutajaliidese raamistikke. Välja saaks tuua Material UI, Semantic UI, React Toolbox'i, Ant Design'i ja React Foundation'i. Autoril oli olemas kogemus React Bootstrap teegi

kasutamisega, kuid polnud mingit kokkupuudet ühegi eelpool nimetatud raamistikuga. Sellepärast otsustas autor edasi liikuda React Bootstrap teegi kasutamisega, kuna selle teegi pakutavad võimalused olid bakalaureusetöö käigus valmiva rakenduse loomiseks piisavad.

React Bootstrap teegist kasutas autor rakenduse päise tegemiseks Navbar komponente. Kuna otsinguvorme on neli, siis on vaikimisi kõik peidus, et neli erinevat otsinguvormi segadust ei tekitaks. Vastava päringu lingile klõpsates avaneb otsinguvorm (joonis 13). Samale lingile uuesti klikkides vorm peidetakse. Mõne teise päringu lingile klõpsates peidetakse avatud vorm ning avatakse uue päringu vorm.



Geodeetiliste punktide andmekogu

Otsi nime järgi Otsi ID järgi Otsi koodnumbri järgi Otsi koordinaadi ümbert Kalkulaatorid ▾

Otsi nime järgi Leia täpne vaste

Joonis 13. Nime järgi otsingu vorm

Geodeetilise punkti detailinfo näitamisel kasutas bakalaureusetöö autor lisaks veel Modal komponenti piltide kuvamise juures. Kus pildid avanesid Modal komponendi sees ning sai edasi-tagasi nuppudega järgmist või eelmist pilti kuvada (joonis 14).



Joonis 14. Rakenduses piltide kuvamise näide

Kalkulaatorite juurde lisas autor võimaluse laadida sisendandmete üleslaadimise võimalus CSV failist. CSV vormingus fail võimaldab tavalises tekstifailis edastada infot tabelina. Esimesel real ehk päises on veergude nimetused ning järgmistel ridadel tulevad kirjed. Veerud eraldatakse reeglina komaga, kuid eraldajaks võidakse kasutada ka mõnda muud sümbolit. [63]

Kui fail oli sisestatud, siis pidi andmed teisendama JSON kujule. Selleks kasutas autor Papa Parse teeki. Proovitud sai ka CSVtoJSON teeki, kuid seal ei olnud võimalik tuvastada automaatselt veergude eraldajat. Lisaks võimaldas Papa Parse teek teisendusi teha mõlemat pidi. Nii CSV'st JSON'iks kui ka vastupidi. Seda läks vaja API'st saadud JSON vastuse tagasi CSV kujule teisendamisel.

Lõpuks sai kasutatud react-download-link nimelist teeki, et luua allalaadimise link, läbi mille kasutaja sai arvutatud tulemustega CSV faili alla laadida (joonis 15).

CSV failist mitmed kirjed korruga

Geo -> L-EST97

L-EST97 -> Geo

CSV fail, veerud: x,y

Vali fail koordinaadid_lest.csv

Käivita

[Laadi tulemus alla](#)

Joonis 15. CSV failist kirjete laadimine ja loodud vastuse link

5.3 Turvalisus

Autor kirjeldab, kuidas rakendati ASVS nõudeid.

5.3.1 Kasutaja sisend

Tagarakenduses kasutatakse andmebaasiga suhtlemisel Spring Boot Starter Data JPA teeki. JPA kaudu tehakse parameetritega JPQL või SQL andmebaasipäringuid, mis tähendab seda, et kasutaja sisestatud väärtuseid ei lisata otse päringu osaks, mis teeks selle SQL süstimise rünnaku suhtes haavatavaks. Päringud luuakse selliselt, et parameetrite jaoks on olemas kohad ning väärtused lisatakse alles vahetult enne päringu käivitamist. [64]

Eesrakenduses näidatakse otsingutulemusel ka otsingu parameetreid. React puhastab väljastatava teksti automaatselt, nii et JSX sees on turvaline kasutaja sisendit kasutada. [65]

Kalkulaatori jaoks on võimalik kasutajatel infot üles laadida failidest. Kalkulaatorite .exe failid loevad vormindatud tekstifaile, milles väljad on kas tühiku või tabulaatoriga eraldatud. Reavahetus alustab uue kirje. Veerud on nimetamata, nii et kasutaja peab ise jälgima, et veergude järjestus oleks õige. Kõige lihtsam olekski olnud kasutada samasuguseid tekstifaile, laadida need API kaudu tagarakendusele kättesaadavaks ning seejärel need kalkulaatorile sisendiks anda. Selle tulemusena oluks vaja teha mitmeid kontrole nii ees- kui ka tagarakenduses, et tegemist ei oleks liiga suure failiga, sisu oleks see, mida eeldatakse. Ka failinimi oleks olnud üks ründevektor.

Bakalaureusetöö autor otsustas, et loeb faili sisse eesrakenduses, teisendab selle JSON vormingusse ning seejärel tehakse POST päring, mille kehas on failist teisendatud JSON objekt, API vastu. Selle jaoks otsustas autor võimaldada eesrakenduses kasutada CSV vormingut. See lisaks juurde paindlikkust kasutajale, kuna veerud on seal juba nimetatud ning ka mõnda aega hiljem faili avades on kohe aru saada, milline veerg sisaldab mis andmeid. Lisaks pole CSV faili puhul veergude järjekord oluline, veerud võivad olla mistahes järjekorras. Tegemist on ka lihtsalt loodava failiga, selle saab luua kasutades kasvõi kõige lihtsamat tekstiredaktorit. Ka tänapäeva tabelarvutuse programmid võimaldavad eksportida infot CSV vormingusse.

JSON objektiks teisendatuna info tagarakendusse üles laadides on võimalik @RequestBody annotatsiooni ja sama struktuuriga klassi kasutades kohe ka kontrollida, kas sissetulev info on õige struktuuri ja andmetüüpidega. Saadetava info maksimaalseks suuruseks POST päringus on Spring Boot'i sisseehitatud Tomcat veebiserveri puhul vaikimisi 2MB. [66] Tellijaga suheldes tuli välja, et sellest peaks piisama, kui vaja, siis saab seda suurust ka suurendada.

Kuna mõned kalkulaatorid võtsid sisendi ainult failidena, siis peab tagarakendusse saadetud JSON vormingus info kirjutama kettale sellises vormingus, nagu kalkulaator seda aktsepteerib. Faili nimi genereeritakse alati juhusliku tähejadana ning peale rakenduse käivitamist sisendfail kustutatakse. Sama on ka väljundfailiga, nimi genereeritakse juhusliku tähejadana ning kui tagarakendus on info sisse lugenud, siis fail kustutatakse. Nii ei jää need failid ka serveri kettale ruumi võtma. Väljundfailist saadud info töödeldakse uuesti objektiks ning loodud objekt läheb API päringu vastuseks.

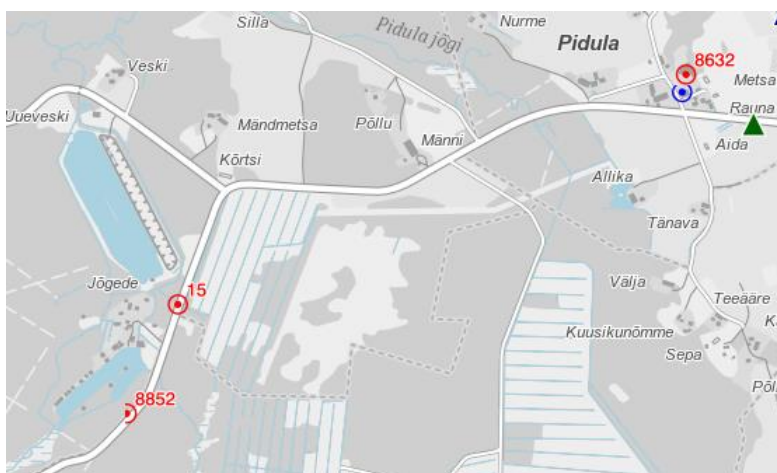
Mõned kalkulaatorid kasutavad ka abifaile sisendiks lisaks kasutaja poolsete andmete sisendfailile. Need failid on juba kalkulaatoritega kaasas, kuid seda, mis faile konkreetselt vaja kasutada on, määrab kasutaja oma eesrakenduse vormis tehtavate valikutega. Siin on oht OS Command Injection rünteks kui failinimi ja kaustatee otse vormist kaasa anda. Sellepärast saadetakse vormist ainult vastava valiku järjekorranumber ning tagarakenduses saadakse vastava numbriga kokkumineva faili asukoht, mis siis kalkulaatori argumendiks lisatakse.

5.3.2 Piltide allalaadimine

Piltide nimedeks failiserveris on pildi ID ja faililaiend. Piltide ID-na aktsepteeritakse ainult Long tüüpi sisendit, et kasutaja ei saaks sisestada mingeid muid andmetüüpe. Sellega kaob võimalus sisestatud ID-sse süstida sõnesid, mille eesmärgiks on süsteemi rünnata.

5.4 Kaardimoodul

Kaardimooduli tegemiseks valis autor välja Leaflet teegi. See tuli lahenduse loomise käigus välja vahetada OpenLayers teegi vastu. Probleemiks oli WMS teenuse puhul tekkinud ebakõlad ruutude liitekohtadel ja märgistusi kuvati poolikult (joonis 16).



Joonis 16. Poolikud märgistused kaardil

Probleem tekkis sellest, kui punkt oli ühe ruudu servas, siis märgistus joonistati samale kaardiruudule, millel punkt oli. Märgistus oleks pidanud jätkuma kõrval oleval ruudul, kuid kuna selle ruudu peal punkti kohta infot ei olnud, siis ei joonistatud sinna midagi ning punkt kaardil jäi poolikuks.

Autor leidis StackExchange foorumist sama probleemi kirjelduse ning seal oli välja toodud kolm lahendust [67]:

1. Ruutude suurendamine;
2. Kogu kihi näitamine ühe ruuduna;
3. Serveri poolse seadistuse muutmise, et pildi joonistamisel kasutataks ka seda infot, mis jääb ruudust väljapoole.

Kuna autoril ei olnud ligipääsu Geoportaali serveri poole, siis kolmas variant jäi kohe kõrvale. Esimese lahenduse puhul oli foorumis välja toodud, et see lahendus ei kaota probleemi, vaid selliseid probleemseid kohti on vähem. Autor proovis selle ise järgi ning katsetus kinnitas, et tõesti, probleemi ilmnes vähem, kuid polnud päris kadunud. Seega jäi järgi teine variant, kogu kihi näitamine ühe ruuduna. Siin tekkiski probleem leaflet teegiga, milles polnud varianti üheruudulise kihti allalaadimiseks. Autor leidis küll kaks kolmanda osapoolte teeki, mis pidanuks seda võimaldama: leaflet.wms ja Leaflet.NonTiledLayer.

Autor ei saanud kumbagi teeki oma React projektis toimima, npm install küll õnnestus, kuid React projektis ei leitud paigaldatud teeki. Kaarditeekide testimise käigus oli kaardimoodul loodud nii OpenLayers kui ka Leaflet teeki kasutades, seega autor ei hakanud väga palju aega vea otsingule kulutama ning võttis kasutusele OpenLayers teegi, kuna seal olid üheruudulised kihid toetatud, kasutades ImageLayer klassi (joonis 17).

```
const markerLayers = [  
  new ImageLayer({  
    extent: MAP_EXTEND_EST,  
    source: new ImageWMS({  
      url: 'https://teenus.maaamet.ee/ows/gpa',  
      params: {  
        'LAYERS': 'RGV,korgusvork,kohalik_vork,gravimeetriline_vork,Seisukorra_teenakaart',  
        'VERSION': '1.1.1'  
      }  
    })  
  })  
];
```

Joonis 17. Kihi lisamine ühe ruuduna

Veel üks probleem tekkis kasutades react-bootstrap Tab komponentides kaardi kasutamisel. Kui kaart oli selles vahelehes nähtaval, mida näidati veebilehe esialgsel laadimisel, siis joonistati pilt korralikult välja. Kui kaart oli mõnel teisel vahelehel, mida kasutaja poolt avati, siis kaardi pilti ei joonistatud. Veebilehitseja akna suuruse muutmisel joonistati ka kaart. See probleem oli mõlema teegiga kui ka X-GIS2 kaardirakenduse kasutamisel iframe akna sees. Selle probleemi lahenduseks oli peale vahelehe avamist välja kutsuda rakenduse siseselt kaardi instantsilt meetod, mis kaardi suurust uuendas, mille peale joonistati kaart. X-GIS2 puhul autor probleemile lahendust ei leidnud.

Leaflet teegil oli selleks meetodiks invalidateSize ning OpenLayers teegil updateSize.

5.5 Kalkulaatorid

Kalkulaatorite algoritmid on välja töötatud Maa-ameti poolt, autor neid välja mõtlema ei pidanud. Nende kalkulaatorite puhul, mis ei kasutanud exe tööriistu, pidi autor PHP koodi Javasse tõstma. Exe tööriistade puhul pidi töötleva programmi poolt standardväljundisse saadetud teksti, milles arvutuste tulemused oli eraldi ridadel ning väärtused eraldatud tühikutega. Selleks kasutas autor regulaaravaldisi, millega saadi väljundist kätte vajalik info ning loodi objektid, mis API kaudu tagastati.

6 Kokkuvõte

Käesoleva bakalaureusetöö käigus teostati analüüs, mille tulemusena selgitati välja sobivaimad tehnoloogiad geodeetiliste punktide andmekogu veebirakenduse loomiseks. Vaadeldi rakenduse turvalisuse kontrollimise standardit ning rakendati sealseid soovitusi rakenduse arenduse juures.

Bakalaureusetöö käigus loodi uue rakenduse esialgne versioon, mis hetkel küll pole veel suuremas kasutuses. Rakendus töötab Keskkonnaministeeriumi Infotehnoloogiakeskuse sisevõrgus olevas serveris ning seda testivad Maa-ameti kasutajad, kelle poolt tulevad muudatus- ja parandusettepanekud. Töö rakenduse kallal veel jätkub vastavalt saadud tagasisidele.

Bakalaureusetöö püstitatud eesmärk sai täidetud. Rakendus võimaldab teha päringuid andmekogusse, näidata detailset info punkti kohta ning selle asukohta kaardil. Kasutatavad on ka geodeesia kalkulaatorid. Kalkulaatorite kasutamiseks valmis ka RESTful API, läbi mille on võimalik sooritada nii ühe kirje arvutusi kui ka massarvutusi kirjeid failist üles laadides.

Rakendus vastab ASVS esimese taseme nõuetele. Turvalisuse testimiseks kasutas autor musta kasti läbistustestimise tööriista Arachni, mille käigus kontrolliti haavatavusi, mida väline kasutaja võiks süsteemi ründamiseks ära kasutada.

Kasutatud kirjandus

- [1] „Geoportaal - Geoid,“ Maa-amet, [Võrgumaterjal]. Available: <https://geoportaal.maaamet.ee/est/Ruumiandmed/Geodeetilised-andmed/Geodeetilised-vorgud/Geoid-p287.html>. [Kasutatud 05 04 2022].
- [2] „Geodeetiliste punktide andmekogu,“ Maa-amet, [Võrgumaterjal]. Available: <https://geoportaal.maaamet.ee/est/Ruumiandmed/Geodeetilised-andmed/Geodeetiliste-punktide-andmekogu-p82.html>. [Kasutatud 21 03 2022].
- [3] „Geodeetiliste punktide andmekogu asutamine ja andmekogu pidamise põhimäärus,“ Riigi Teataja, 22 12 2011. [Võrgumaterjal]. Available: <https://www.riigiteataja.ee/akt/127122011006>. [Kasutatud 21 03 2022].
- [4] „Ruumiandmete seadus,“ Riigi Teataja, 10 03 2011. [Võrgumaterjal]. Available: <https://www.riigiteataja.ee/akt/128022011002?leiaKehtiv>. [Kasutatud 21 03 2022].
- [5] „Geodeesia kalkulaatorid,“ Maa-amet, [Võrgumaterjal]. Available: <https://geoportaal.maaamet.ee/est/teenused/geodeesia-kalkulaatorid-p142.html>. [Kasutatud 22 03 2022].
- [6] „KEM haldusala tehnoloogiline profiil,“ KeMIT, 18 11 2021. [Võrgumaterjal]. Available: <https://www.kemit.ee/sites/kemit/files/2022-02/KEM-haldusala-tehnoloogiline-profiil-2021-11-18.pdf>. [Kasutatud 06 03 2022].
- [7] „Frontend vs Backend,“ 31 08 2021. [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/frontend-vs-backend/>. [Kasutatud 22 04 2022].
- [8] Rollbar Editorial Team, „10 Best PHP Frameworks For Savvy Web Devs In 2022,“ Rollbar, 08 10 2021. [Võrgumaterjal]. Available: <https://rollbar.com/blog/10-best-php-frameworks/>. [Kasutatud 06 03 2022].
- [9] „Spring vs. Spring Boot vs. Spring MVC,“ Java T Point, [Võrgumaterjal]. Available: <https://www.javatpoint.com/spring-vs-spring-boot-vs-spring-mvc>. [Kasutatud 15 03 2022].
- [10] „Spring Boot,“ VMware, Inc., [Võrgumaterjal]. Available: <https://spring.io/projects/spring-boot>. [Kasutatud 20 03 2022].
- [11] „Pros and Cons of Using Spring Boot,“ Bamboo agile, 04 07 2021. [Võrgumaterjal]. Available: <https://bambooagile.eu/insights/pros-and-cons-of-using-spring-boot/>. [Kasutatud 20 03 2022].
- [12] „What is Django?,“ PythonForBeginners.com, [Võrgumaterjal]. Available: <https://www.pythonforbeginners.com/learn-python/what-is-django>. [Kasutatud 21 03 2022].
- [13] J. Korsun, „Why We Use Django Framework & What Is Django Used For,“ Django Stars, 25 01 2016. [Võrgumaterjal]. Available: <https://djangostars.com/blog/why-we-use-django-framework/>. [Kasutatud 21 03 2022].

- [14] „Django introduction,“ Mozilla Corporation, 18 02 2022. [Võrgumaterjal]. Available: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>. [Kasutatud 21 03 2022].
- [15] „Django Advantages and Disadvantages – Why You Should Choose Django?,“ DataFlair, [Võrgumaterjal]. Available: <https://data-flair.training/blogs/django-advantages-and-disadvantages/>. [Kasutatud 21 03 2022].
- [16] „Why to Use Django Web Framework for Web Development Projects?,“ Monocubed, 15 11 2021. [Võrgumaterjal]. Available: <https://www.monocubed.com/blog/django-web-framework/>. [Kasutatud 21 04 2022].
- [17] „Laravel Introduction,“ Laravel LLC, [Võrgumaterjal]. Available: <https://laravel.com/docs/4.2/introduction>. [Kasutatud 21 03 2022].
- [18] „A brief guide through Laravel,“ mDevelopers, 12 01 2022. [Võrgumaterjal]. Available: <https://mdevelopers.com/blog/a-brief-guide-through-laravel>. [Kasutatud 21 03 2022].
- [19] „Artisan Console,“ Laravel LLC, [Võrgumaterjal]. Available: <https://laravel.com/docs/9.x/artisan>. [Kasutatud 21 03 2022].
- [20] S. Jaiswal, „What is Laravel? Advantages & Disadvantages,“ 17 11 2019. [Võrgumaterjal]. Available: <https://www.courseya.com/blog/what-is-laravel-features-and-advantages/>. [Kasutatud 02 04 2022].
- [21] M. Bhatnagar, „Magic lies here - Statically vs Dynamically Typed Languages,“ [Võrgumaterjal]. Available: <https://medium.com/android-news/magic-lies-here-statically-typed-vs-dynamically-typed-languages-d151c7f95e2b>. [Kasutatud 02 04 2022].
- [22] „Spring Boot GitHub Repository,“ [Võrgumaterjal]. Available: <https://github.com/spring-projects/spring-boot>. [Kasutatud 02 04 2022].
- [23] „Laravel GitHub repository,“ [Võrgumaterjal]. Available: <https://github.com/laravel/laravel>. [Kasutatud 02 04 2022].
- [24] Stackoverflow, „Stackoverflow annual developer survey 2021,“ 2021. [Võrgumaterjal]. Available: <https://insights.stackoverflow.com/survey/2021#section-most-popular-technologies-web-frameworks>. [Kasutatud 09 03 2022].
- [25] „What is Angular,“ Google, 28 10 2021. [Võrgumaterjal]. Available: <https://angular.io/guide/what-is-angular>. [Kasutatud 15 03 2022].
- [26] „The Good and the Bad of Angular Development,“ AltexSoft, 25 03 2020. [Võrgumaterjal]. Available: <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-angular-development/>. [Kasutatud 15 03 2022].
- [27] „Dependency injection in Angular,“ Google, [Võrgumaterjal]. Available: <https://angular.io/guide/dependency-injection>. [Kasutatud 15 03 2022].
- [28] A. Gazta, „Understanding the pros and cons of Angular Development,“ GreyCampus, 12 10 2021. [Võrgumaterjal]. Available: <https://www.greycampus.com/blog/programming/good-and-bad-of-angular-development>. [Kasutatud 15 03 2022].
- [29] „Advantages and Disadvantages of Angular,“ KnowledgeHut, 15 03 2021. [Võrgumaterjal]. Available: <https://www.knowledgehut.com/blog/web-development/advantages-and-disadvantages-of-angular>. [Kasutatud 15 03 2022].

- [30] K. Sheth, „What is Virtual DOM? How Virtual DOM works ? What is Reconciliation ? What is diffing algorithm? What makes React so fast ?“, DEV, 27 04 2021. [Võrgumaterjal]. Available: <https://dev.to/koolkishan/what-is-virtual-dom-how-virtual-dom-works-what-is-reconciliation-what-is-diffing-algorithm-what-makes-react-so-fast-327a>. [Kasutatud 09 03 2022].
- [31] D. Shapiro, „Understanding Component-Based Architecture“, 16 06 2016. [Võrgumaterjal]. Available: <https://medium.com/@dan.shapiro1210/understanding-component-based-architecture-3ff48ec0c238>. [Kasutatud 02 04 2022].
- [32] R. Barger, „Is React a Library or a Framework? Here's Why it Matters“, freeCodeCamp, 12 04 2021. [Võrgumaterjal]. Available: <https://www.freecodecamp.org/news/is-react-a-library-or-a-framework/>. [Kasutatud 09 03 2022].
- [33] H. Dhaduk, „Angular vs React 2022 : Which JS Framework your Project Requires?“, 16 02 2021. [Võrgumaterjal]. Available: <https://www.simform.com/blog/angular-vs-react/>. [Kasutatud 03 04 2022].
- [34] „GitHub - angular/angular: The modern web developer's platform“, [Võrgumaterjal]. Available: <https://github.com/angular/angular>. [Kasutatud 03 04 2022].
- [35] „GitHub - facebook/react: A declarative, efficient, and flexible JavaScript library for building user interfaces.“, [Võrgumaterjal]. Available: <https://github.com/facebook/react>. [Kasutatud 03 04 2022].
- [36] Angular University, „Angular Change Detection - How Does It Really Work?“, 21 01 2022. [Võrgumaterjal]. Available: <https://blog.angular-university.io/how-does-angular-2-change-detection-really-work/>. [Kasutatud 03 04 2022].
- [37] L. Gupta, „What is REST“, 07 04 2022. [Võrgumaterjal]. Available: <https://restfulapi.net/>. [Kasutatud 06 05 2022].
- [38] M. Richards, Software Architecture Patterns, O'Reilly Media Inc, 2015.
- [39] „OWASP Application Security Verification Standard - What is ASVS?“, The OWASP® Foundation, [Võrgumaterjal]. Available: <https://owasp.org/www-project-application-security-verification-standard/>. [Kasutatud 22 03 2022].
- [40] The OWASP Foundation, „Application Security Verification Standard 4.0.3“, 10 2021. [Võrgumaterjal]. Available: <https://github.com/OWASP/ASVS/blob/v4.0.3/4.0/OWASP%20Application%20Security%20Verification%20Standard%204.0.3-en.pdf>. [Kasutatud 22 03 2022].
- [41] „Command Injection“, Imperva, [Võrgumaterjal]. Available: <https://www.imperva.com/learn/application-security/command-injection/>. [Kasutatud 19 04 2022].
- [42] „WSTG - v4.2 - Introduction“, OWASP, [Võrgumaterjal]. Available: <https://owasp.org/www-project-web-security-testing-guide/v42/2-Introduction/>. [Kasutatud 29 03 2022].
- [43] V. Agafonkin, „Leaflet“, [Võrgumaterjal]. Available: <https://leafletjs.com/SlavaUkraini/>. [Kasutatud 29 03 2022].
- [44] V. Agafonkin, „Leaflet - Projection“, [Võrgumaterjal]. Available: <https://leafletjs.com/SlavaUkraini/reference.html#projection>. [Kasutatud 29 03 2022].

- [45] Teadmata, „Kaardiprojektsioon,“ Tartu Ülikool, [Võrgumaterjal]. Available: <http://www.geo.ut.ee/kooligeo/EGCD/opik/juts/karto/projekts.html>. [Kasutatud 29 03 2022].
- [46] „EPSG:4326,“ EPSG.io, 27 08 2007. [Võrgumaterjal]. Available: <https://epsg.io/4326>. [Kasutatud 30 03 2022].
- [47] W. MathWorld, „Cylindrical Projection,“ [Võrgumaterjal]. Available: <https://mathworld.wolfram.com/CylindricalProjection.html>. [Kasutatud 29 03 2022].
- [48] „EPSG:3395,“ EPSG.io, 02 06 2006. [Võrgumaterjal]. Available: <https://epsg.io/3395>. [Kasutatud 30 03 2022].
- [49] „Leaflet vs OpenLayers,“ Geoapify, 12 04 2019. [Võrgumaterjal]. Available: <https://www.geoapify.com/leaflet-vs-openlayers>. [Kasutatud 30 03 2022].
- [50] „OpenLayers,“ [Võrgumaterjal]. Available: <https://openlayers.org/>. [Kasutatud 30 03 2022].
- [51] „X-GIS 2 kasutajaliides,“ Maa-amet, [Võrgumaterjal]. Available: <https://geoportaal.maaamet.ee/est/Teenused/X-GIS-2-kasutajaliides-p675.html>. [Kasutatud 03 04 2022].
- [52] „WMS/WFS teenused,“ Maa-amet, 01 03 2022. [Võrgumaterjal]. Available: <https://geoportaal.maaamet.ee/est/Teenused/WMSWFS-teenused-p65.html>. [Kasutatud 23 04 2022].
- [53] „WMS reference,“ Open Source Geospatial Foundation, [Võrgumaterjal]. Available: <https://docs.geoserver.org/stable/en/user/services/wms/reference.html>. [Kasutatud 23 04 2022].
- [54] „WMS teenuse kasutusjuhendid,“ Maa-amet, 10 02 2020. [Võrgumaterjal]. Available: <https://geoportaal.maaamet.ee/est/Teenused/WMS-teenused/WMS-teenuse-kasutusjuhendid-p122.html>. [Kasutatud 23 04 2022].
- [55] „Maa-ameti WMS teenuste kasutamine,“ Maa-amet, 2019. [Võrgumaterjal]. Available: https://geoportaal.maaamet.ee/docs/WMS/API_juhend_MA_2019.pdf?t=20190503124448. [Kasutatud 03 04 2022].
- [56] A. Kim, „react-openlayers,“ [Võrgumaterjal]. Available: <https://github.com/allenhwkim/react-openlayers>. [Kasutatud 19 04 2022].
- [57] „Java Archive (JAR) Files,“ Oracle, [Võrgumaterjal]. Available: <https://docs.oracle.com/javase/6/docs/technotes/guides/jar/index.html>. [Kasutatud 10 04 2022].
- [58] „Stackoverflow - Cannot run exe after building the jar,“ 2013. [Võrgumaterjal]. Available: <https://stackoverflow.com/questions/19299470/cannot-run-exe-after-building-the-jar>. [Kasutatud 10 04 2022].
- [59] D. G. Pelzel, „Why "optimizing" your images with Base64 is almost always a bad idea,“ 17 03 2019. [Võrgumaterjal]. Available: <https://bunny.net/blog/why-optimizing-your-images-with-base64-is-almost-always-a-bad-idea/>. [Kasutatud 04 04 2022].
- [60] Baeldung, „How Can I Resize an Image Using Java?,“ 22 05 2021. [Võrgumaterjal]. Available: <https://www.baeldung.com/java-resize-image>. [Kasutatud 04 04 2022].
- [61] E. A, M. S ja O. T, „Eesti geoidi mudel EST-GEOID2017,“ Tallinna Tehnikaülikool, 2017.

- [62] „React Bootstrap,“ [Võrgumaterjal]. Available: <https://react-bootstrap.github.io>. [Kasutatud 10 04 2022].
- [63] C. Hoffman, „What Is a CSV File, and How Do I Open It?,“ 18 04 2018. [Võrgumaterjal]. Available: <https://www.howtogeek.com/348960/what-is-a-csv-file-and-how-do-i-open-it/>. [Kasutatud 20 04 2022].
- [64] T. Janssen, „Native Queries – How to call native SQL queries with JPA & Hibernate,“ [Võrgumaterjal]. Available: <https://thorben-janssen.com/jpa-native-queries/>. [Kasutatud 19 04 2022].
- [65] „Introducing JSX,“ Meta Platforms, Inc, [Võrgumaterjal]. Available: <https://reactjs.org/docs/introducing-jsx.html>. [Kasutatud 19 04 2022].
- [66] P. Gupta, „Tomcat Server - Request Size,“ 07 06 2020. [Võrgumaterjal]. Available: <https://medium.com/@priyammgupta/tomcat-server-request-size-d5e8807a990c>. [Kasutatud 20 04 2022].
- [67] „Preventing clipped labels from QGIS Server WMS in Leaflet JS,“ StackExchange, [Võrgumaterjal]. Available: <https://gis.stackexchange.com/questions/346467/preventing-clipped-labels-from-qgis-server-wms-in-leaflet-js>. [Kasutatud 23 04 2022].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Riho Kalda

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Geodeetiliste punktide andmekogu turvalise veebirakenduse arendus“, mille juhendaja on Aleksei Talisainen
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

15.05.2022

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Kohalike koordinaadisüsteemide loend

- Harjumaa - Aegna
- Harjumaa - Aegviidu
- Harjumaa - Aruküla
- Harjumaa - Jüri
- Harjumaa - Kehra
- Harjumaa - Keila
- Harjumaa - Kose
- Harjumaa - Kostivere
- Harjumaa - Lagedi
- Harjumaa - Loksa
- Harjumaa - Maardu
- Harjumaa - Our
- Harjumaa - Paldiski
- Harjumaa - Raasiku
- Harjumaa - Saku
- Harjumaa - Saue
- Harjumaa - Tabasalu
- Harjumaa - Tallinn
- Harjumaa - Vaida
- Harjumaa - Vasalemma
- Harjumaa - Ääsmäe
- Hiiumaa - Käina
- Hiiumaa Kärddla
- Ida-Virumaa - Ahtme
- Ida-Virumaa - Alajõe
- Ida-Virumaa - Aseri
- Ida-Virumaa - Iisaku
- Ida-Virumaa - Jõhvi
- Ida-Virumaa - Kaevanduse (Eesti Põlevkivi 1973.a.)
- Ida-Virumaa - Karjamaa-Smolnitsa
- Ida-Virumaa - Kiviõli

- Ida-Virumaa - Kohtla-Järve - Ahtme linnaosa
- Ida-Virumaa - Kohtla-Järve - Järve linnaosa
- Ida-Virumaa - Kohtla-Järve - Sompa linnaosa
- Ida-Virumaa - Kurtna
- Ida-Virumaa - Mäetaguse
- Ida-Virumaa - Narva
- Ida-Virumaa - Narva-Jõesuu
- Ida-Virumaa - Oru
- Ida-Virumaa - Püssi
- Ida-Virumaa - Sillamäe
- Ida-Virumaa - Sonda
- Ida-Virumaa - Toila
- Ida-Virumaa - Voka
- Järvamaa - Ambla
- Järvamaa - Aravete
- Järvamaa - Imavere
- Järvamaa - Järva-Jaani
- Järvamaa - Koeru
- Järvamaa - Oisu
- Järvamaa - Paide
- Järvamaa - Peetri
- Järvamaa - Roosna-Alliku
- Järvamaa - Türi
- Järvamaa - Väätsa
- Jõgevamaa - Jõgeva
- Jõgevamaa - Mustvee
- Jõgevamaa - Pisisaare
- Jõgevamaa - Põltsamaa
- Jõgevamaa - Torma
- Lääne-Virumaa - Haljala
- Lääne-Virumaa - Kadrina
- Lääne-Virumaa - Kunda
- Lääne-Virumaa - Laekvere
- Lääne-Virumaa - Rakke

- Lääne-Virumaa - Rakvere
- Lääne-Virumaa - Simuna
- Lääne-Virumaa - Tamsalu
- Lääne-Virumaa - Tapa
- Lääne-Virumaa - Toolse piirkond
- Lääne-Virumaa - Väike-Maarja
- Lääne-Virumaa - Võsu
- Lääne-Virumaa - Võsu 42
- Läänemaa - Haapsalu_01
- Läänemaa - Haapsalu_02
- Läänemaa - Haapsalu_03
- Läänemaa - Palivere
- Läänemaa - Risti
- Läänemaa - Taebla
- Läänemaa - Uuemõisa
- Läänemaa - Virtsu
- Pärnumaa - Are
- Pärnumaa - Audru
- Pärnumaa - Häädemeeste
- Pärnumaa - Kilingi-Nõmme
- Pärnumaa - Lavassaare
- Pärnumaa - Lihula
- Pärnumaa - Paikuse
- Pärnumaa - Pärnu
- Pärnumaa - Pärnujaagupi
- Pärnumaa - Sindi
- Pärnumaa - Tootsi
- Pärnumaa - Tori
- Pärnumaa - Tõstamaa
- Pärnumaa - Vändra
- Pärnumaa - Vändra EP
- Põlvamaa - Põlva
- Põlvamaa - Räpina
- Raplamaa - Juuru

- Raplamaa - Kohila
- Raplamaa - Käru
- Raplamaa - Rapla
- Raplamaa - Vana-Vigala
- Saaremaa - Kuresaare
- Saaremaa - Leisi
- Saaremaa - Orissaare
- Tartumaa - Alatskivi
- Tartumaa - Elva
- Tartumaa - Kallaste
- Tartumaa - Karkna
- Tartumaa - Kõrvekula
- Tartumaa - Lähte
- Tartumaa - Mehikoorma
- Tartumaa - Nõo
- Tartumaa - Puhja
- Tartumaa - Rannu
- Tartumaa - Rõngu
- Tartumaa - Tabivere
- Tartumaa - Tartu
- Tartumaa - Vara
- Valgamaa - Otepää
- Valgamaa - Puka
- Valgamaa - Sihva
- Valgamaa - Tsirguliina
- Valgamaa - Tõrva
- Valgamaa - Valga
- Viljandimaa - Abja
- Viljandimaa - Kõpu
- Viljandimaa - Mustla
- Viljandimaa - Mõisaküla
- Viljandimaa - Nuia
- Viljandimaa - Saarepeedi
- Viljandimaa - Suure-Jaani

- Viljandimaa - Vastsemõisa
- Viljandimaa - Viljandi
- Viljandimaa - Võhma
- Võrumaa - Antsla
- Võrumaa - Misso
- Võrumaa - Vastseliina
- Võrumaa - Viitina
- Võrumaa - Võru