

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Andreas Porman 175874IDDR

Veekogude põhja seire ja kaardistuse platvormi prototüüp

Diplomitöö

Juhendaja: Toomas Lepikult
Agu Leinfeld

Tallinn 2021

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: Andreas Porman

28.04.2021

Annotatsioon

Käesolev diplomitöö käsitleb veekogude seire ja põhjakaardi tootmise võimekuse loomist ning aitab ettevõttel AS Datel otsustada uue tootearenduse ja ärisuuna avamise. Töö käigus luuakse prototüüplahendus, millega rahuldatakse kliendi etteantud ärinõudeid veekogude põhjakaardi tootmise platvormile. Prototüübis töödeldakse sonariga kogutud toorandmed ruumiandmete kujule, mida on võimalik kasutada nii sügavuskaartide, põhja reljeefi 3D kui ka kõrgusmodelite tootmiseks ning loodud tulemuste abil aidata vastu võtta ärilisi otsuseid. Diplomitöö eesmärk on koostada tehnilise teostatavuse uuring ja prototüüpimine tootearenduse idee „veekogude põhjakaardi tootmise platvormi“ kuni TRL 5 tasemel kohta.

Diplomitöö teoreetilises osas valis ja põhjendas autor kasutatuid meetodikaid ja andis ülevaate valdkonnast ning tutvustas põgusalt juba turul olevaid sarnaseid tooteid ja teenuseid.

Diplomitöö praktilises osas tutvustab autor loodava toote disaini protsessi ja kirjeldab prototüüpi ning selle loomist. Disaini protsessis toob autor välja nõuete kogumise ja prioritseerimise kirjelduse ning prototüübi loomiseks valitud tehnoloogiad ja tarkvara arenduse mustrid. Prototüübi kirjeldamise peatükis toob autor välja prototüüplahenduse erinevad osad ja nende loomise kirjelduse ning probleemset kohad.

Töö peamised järeldused on esitatud diplomitöö arutelu ja järelduste peatükis.

Diplomitöö on kirjutatud eesti keeles ja sisaldab 65 lehekülge, 7 peatükki, 13 joonist ning 3 tabelit. Diplomitöö annotatsioon on esitatud ka inglise keeles.

Abstract

Monitoring and Mapping Platform Prototype for Water Bodies

The problem of the diploma's thesis was that company AS Datel created space based infrastructure deformation monitoring platform called SILLE. While SILLE covers the entire world, it does not „see“ under the water.

The thesis of this diploma deals with the creation of waterbody monitoring and bottom map production platform and helps the company AS Datel to decide whether to start a new product development and open a new business direction. In the course of the work, a prototype solution will be created, which will satisfy the business requirements set by the customer for the bottom map production platform. The prototype processes raw data collected by sonar into spatial data that can be used to produce depth maps, 3D terrain and elevation models, and use the results to help to make business decisions. The aim of the diploma thesis is to prepare a technical feasibility study and prototyping of the product development idea “ monitoring and mapping platform for water bodies” up to TRL level 5.

In the theoretical part of the diploma thesis, the author chose and explained the used methodologies and gave an overview of the field and briefly introduced similar products and services already on the market.

In the practical part of the diploma thesis, the author introduces the design process of the created product and describes the prototype and its creation. In the design process, the author presents a description of the collection and prioritization of requirements and the technologies and patterns of software development chosen to create the prototype. In the chapter describing the prototype, the author points out the different parts of the prototype solution and the description of their creation, as well as the arisen problems.

The main conclusions of the dissertation are presented in the discussion and conclusions chapter of the dissertation.

The diploma thesis is written in Estonian and contains 65 pages, 7 chapters, 13 figures and 3 tables.

Sisukord

Autorideklaratsioon.....	2
Annotatsioon.....	3
Abstract.....	4
Sisukord	5
Lühendite ja mõistete sõnastik.....	8
Jooniste loetelu.....	9
Tabelite loetelu.....	10
Sissejuhatus.....	11
1. Probleemi kirjeldus ja ülesande püstitus	12
1.1. Probleemi kirjeldus ja eesmärk	12
1.2. Valdkonna ülevaade	13
1.3. Töö skoop.....	13
1.4. Autori roll.....	14
1.5. Töö struktuur	14
2. Metoodika ja analüüs.....	16
2.1. Arendusmetoodika ja nõuete kogumise tehnikate valiku põhjendus	16
2.2. Valitud projekti arendusmetoodika	16
2.2.1. MoSCoW	17
2.2.2. WBS.....	18
2.3. Nõuete kogumine	18
2.3.1. Lean Canvas.....	19
2.3.2. Requirements Discovery Canvas	20
2.3.3. MVP.....	20
2.3.4. Liftikõne.....	20
3. Olemasolevate lahenduste analüüs	21

3.1.	Äriülesande püstitus	21
3.2.	Turul olevate teenuste/toodete analüüs	22
3.2.1.	Bio Base Maps	22
3.2.2.	Garmin ActiveCaptain	23
3.2.3.	R-Süsteemid RAN, AEGIR ja HIS	24
3.3.	Konkureerivate toodete võrdlus	24
4.	Disaini protsessi kirjeldus.....	26
4.1.	Nõuete kogumise ja prioritseerimise tööprotsess.....	26
4.2.	Valitud tehnoloogiad	28
4.3.	Valitud tarkvara arenduse mustrid	30
4.4.	Rakenduse arhitektuur.....	32
4.5.	Rakenduse andmemudeli kasutamine	33
5.	Prototüübi kirjeldus	35
5.1.	Projekti püstipanek.....	35
5.2.	Andmed	36
5.3.	GIS-lahendused	37
5.4.	Serverrakendus	40
5.5.	Klientrakendus	41
6.	Arutelu ja järeldused.....	43
6.1.	Püstitatud eesmärkide täitmine.....	43
6.2.	Töö rakendamisest töökohal.....	45
	Kokkuvõte.....	46
	Summary	48
	Kasutatud kirjandus	50
	Lisa 1 - Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks.....	53
	Lisa 2 – WBS	54

Lisa 3 – ITIL-i Dateli protsessid teenuste elutsükliga	55
Lisa 4 – TRL tasemed (Bolat, 2014).....	56
Lisa 5 – Lean ja Requirements Discovery Canvas näited.....	57
Lisa 6 – Tootearendusfaasi esimesed eepikud	58
Lisa 7 – MoSCoW meetodil prioritseeritud lähteülesanne	60
Lisa 8 – Võrdlus turulolevate konkureerivate toodete/teenustega	61
Lisa 9 – LAINE kommunikatsiooni arhitektuur	62
Lisa 10 – C-Map Genesis sügavuskaart (kuvatõmmis C-Map Genesis)	63
Lisa 11 – LAINE ERD mudel.....	64
Lisa 12 – Sonarist saadud toorandmed	65

Lühendite ja mõistete sõnastik

DEM	Digital Elevation Model. Digitaalne kõrgusmudel.
GIS	A Geographic Information System. Geoinfosüsteem ehk kohateabesüsteem.
GRASS	Geographic Resources Analysis Support System (GRASS GIS). Geoinfosüsteemi tarkvara.
Evitusskeem	Näitab füüsilisi seoseid tarkvara ja riistvara komponentide vahel.
ITIL	Information Technology Infrastructure Library. Infotehnoloogia infrastruktuuride loetelu.
LAINÉ	Käesolevas töös käsitletav platvorm. Infosüsteem.
MoSCoW	Akronüüm prioritseerimise kategooriatest (Must have, Should have, Could have, Won't have). Lihtsama hääldamise jaoks on lisatud o-tähed.
MVC	Model-View-Controller. Tarkvara arhitektuurimuster.
MVP	A Minimum Viable Product. Minimaalne töötav toode.
POC	Proof of Concept. Minimaalse kasuliku funktsionaalsusega töötav toode tõestamaks idee võimalikkust.
Prototüüp	Kontseptsiooni või protsessi testimiseks ehitatud esialgne proovtoode, mudel või versioon.
Polügoon	Vektorandmete üks geomeetria tüüpe. Joonte abil piiritletud pind kui 2D ruumiobjekt.
REST	Representational State Transfer. Tarkvaraarhitektuuri laad.
TRL	Technology Readiness Levels. Tehnoloogia valmidusastmed. Kontseptsioon, mis võimaldab hinnata arendatava tehnoloogia küpsust.
Sonar	Sound Navigation and Ranging. Tehnoloogia, mis võimaldab heli abil navigeerida ja (objekti) kaugust määrata.
WBS	A Work-breakdown Structure. Tööde hierarhiline struktuur. Protsess, mille käigus jagatakse terve projekt meeskonna vahel paljudeks lihtsamini juhitavateks osadeks.
WFS	Web Feature Service. OGC WFS standardil põhinev kaarditeenus, mis võimaldab andmete kasutamist kolmandates süsteemides.
WMS	Web Map Service. OGC WMS standardil põhinev kaarditeenus, mis võimaldab aluskaartide kasutamist kolmandates süsteemides.

Jooniste loetelu

Joonis 1. Bio Base Map (kuvatõmmis Bio Base Maps rakendusest).....	23
Joonis 2. ActiveCaptain (kuvatõmmis ActiveCaptain rakendusest).....	24
Joonis 3. MVC diagramm (MDN Web Docs)	30
Joonis 4. Domain Model (Fowler, 2003)	32
Joonis 5. Data Mapper (Fowler, 2003)	32
Joonis 6. Evitusskeem.....	33
Joonis 7. ERD diagramm	34
Joonis 8. Esri shape fail koos atribuutandmete tabeliga	38
Joonis 9. Rõuge järve põhi 3D.....	39
Joonis 10. LAINE projekti struktuur	40
Joonis 11. RESTful POST päring	41
Joonis 12. LAINE sügavusandmed.....	41
Joonis 13. Aluskaardi seadistamine L-EST97 koordinaatsüsteemi	42

Tabelite loetelu

Tabel 1. Süsteemi / tarkvaratoote kvaliteet (EVS-ISO/IEC 25010:2011põhjal)(Tepandi, 2020).	19
Tabel 2. Konkureerivate toodete võrdlus.....	25
Tabel 3. Lähteülesandes esitatud funktsionaalsete ärinõuete rahuldamise võrdlus.....	44

Sissejuhatus

Tänapäeva info-ajastul, kui üha rohkematele seadeldistele pannakse külge andureid, mis võimaldavad üha enam ja enam meid ümbritseva keskkonna kohta infot koguda, on sama tähtis ühtlasi ka sellest infost kasulik ja vajalik välja sõeluda.

Kaasaegne kaugseire annab meile uskumatu detailsuse ja täpsusega infot looduslike ja inimese loodud maastike kohta. Tänu erinevatele kaugseire lahendustele saame infot maapinna kohta hõlpsalt kätte nii toorandmete kui ka juba valmis analüüside põhjal. On siiski üks oluline osa maastikust, mille kohta tänapäeva kaugseire kas ei saa infot või on selle info kättesaamine kulukas ja logistiliselt keeruline. Selleks on võimekus näha vee alla ja oskus sügavusandmestikust huvipakkuv ja oluline kätte saada ning tarbitavaks muuta.

Käesolev diplomitöö käsitleb veekogude seire ja põhjakaardi tootmise võimekuse loomist ning aitab ettevõttel AS Datel otsustada uue tootearenduse ja ärisuuna avamise. Sonariga kogutud andmestikust on võimalik kätte saada nii veekogu sügavusinfot kui ka andmeid veekogu põhja koostise ja taimedega kaetuse kohta. Töö käigus luuakse prototüüplahendus, millega rahuldatakse kliendi etteantud ärinõudeid veekogude põhjakaardi tootmise platvormile. Prototüübis töödeldakse sonariga kogutud toorandmed ruumiandmete kujule, mida on võimalik kasutada nii sügavuskaartide, põhja reljeefi 3D kui ka kõrgusmodelite tootmiseks ning loodud tulemuste abil aidata vastu võtta ärilisi otsuseid.

Diplomitöö eesmärk on koostada tehnilise teostatavuse uuring ja prototüüpimine tootearenduse idee „veekogude põhjakaardi tootmise platvormi“ kuni TRL 5 tasemel kohta.

1. Probleemi kirjeldus ja ülesande püstitus

Käesolevas peatükis annab töö autor ülevaate lahendatavast probleemist, käsitletavast valdkonnast ja diplomitöö raames püstitatud eesmärgist.

1.1. Probleemi kirjeldus ja eesmärk

Ettevõtte AS Datel on loonud maapealsete objektide varajase hoiatussüsteemi SILLE, mille abil on võimalik anda objektile varajane hoiatus või määrata objekti riski taset. Maakera on 29.2% ulatuses kaetud maismaaga (Earth), millest SILLE suudab katta 100%. Paraku vee alla SILLE ei näe. Vee alla nägemiseks kaalub AS Datel avada uue ärisuuna ning on käivitanud selleks tootearendusprojekti nimega LAINE. Käesolev töö aitab teha ärilisi otsuseid antud valdkonna jätkusuutlikkuse osas.

Veekogude põhja kaardistus ja põhja profiili ajas muutumine on oluline mitmes valdkonnas. Seda on võimalik rakendada mitmes sektoris:

- erasektor (planeerimine, seire);
- riigisektor (järelevalve, välitööd, keskkond).

Ettevõtte on eraldanud ka tööaega kolmele inimesele prototüübi loomiseks ning lisaks käesolevale tööle on antud teemast valmimas ka magistritöö.

Lõputöö eesmärk on luua ettevõtte siseveekogu põhjakaardi koostamise platvormi prototüüp (TRL 5) (vt TRL tasemeid Lisas 4). Platvormil on kaks peamist ülesannet:

- luua veekogu põhjast aluskaart;
- jäädvustada veekogu põhja profiil.

Uue toote eelis on baasplatvormi universaalsus: platvorm ei ole seotud ühegi konkreetse mehitemata seadmega ega ühegi konkreetse töö ülesandega.

EESMÄRK: Tootearenduse idee „veekogude põhjakaardi tootmise platvormi“ tehnilise teostatavuse uuring ja prototüüpimine kuni TRL 5 tasemel.

Tootearenduse idee valideerimise eesmärk on leida vastused küsimustele kas ettevõttel on mõttekas tootearendusega jätkata ja kas ning millist väärtust see loob potentsiaalsetele kasutajatele.

1.2. Valdkonna ülevaade

Kaugseire tähendab eemal asuvate objektide kohta informatsiooni hankimist mittekontaktsete meetoditega. Kaugseire elektromagnetkiirguse mõõtmisi teostatakse nii satelliitidelt kui lennukitelt või mõõdetakse sonariga veesõidukitelt. Kaugseire töötab pööratud probleemi põhimõttel. Kui huvipakkuvat objekti või nähtust ei saa otse mõõta, on sageli võimalik mõõta mingeid muid sellega seotud objekte ja nähtusi ning huvipakkuva objekti kohta saadakse andmeid mõõdetud andmeid töödeldes. Kaugseire lahutusvõime mõjutab andmete kogumist. Väiksema lahutusvõimega saame vähem detaile ja suurema katvuse, suurema lahutusvõimega rohkem detaile ja väiksema katvuse. Andmete kogumisel tuleb vältida niisuguseid olukordi, kus suure lahutusvõimega uuringute andmed kipuvad ülekande- ja mäluseadmeid ummistama. Kaugseire andmete kvaliteet sõltub ruumilisest, spektraalsest, radiomeetrilisest ja ajalisest lahutusvõimest. (Kaugseire, 2020)

Ruumiline lahutus on piksli suurus, mis on salvestatud rasterpilti. Tavaliselt vastavad pikslid ruudule, mille küljepikkus on 1–1000 meetrit.

Veealuste andmete kogumiseks kasutatakse põhiliselt sonareid. Sonareid on olemas kahte tüüpi: 1) passiivne sonar, mis võtab ainult teiste laevade/objektide poolt tehtavat heli vastu; 2) aktiivne sonar, mis saadab välja heli-impulsse ja võtab vastu nii kaja kui ka teiste objektide poolt tehtavat heli (Simmonds & MacLennan, 2005). Sonar mõõdab kaja ulatuse alusel objekti kauguse ühikuliselt vee all. On teada, et helilained liiguvad vees kiirusega üks miil sekundis. Põhinedes sellele seaduspärasusele ja füüsika reeglitele, arvutab sonar selle aja kulumise järgi, millal heli levib objektini ja tagasi, objekti kauguse. Foote (2008) järgi saab sonarit edukalt kasutada veealusteks kaardistustöödeks ja fotode tegemiseks, veekogude mõõdistamiseks, bioloogiliste organismide jälgimiseks ning objektide järelevalveks ja jälgimiseks.

1.3. Töö skoop

Projekti eesmärk on luua toode, mis annab ettevõttele võimekuse pakkuda potentsiaalsetele klientidele veekogus mõõdetud andmetest terviklikke ajas uuenevatele andmekomplektidele

tuginevaid teenuseid. Antud lõputöö skoopi jääb prototüüplahenduse loomine, mis võimaldab juba olemasolevatest sonariga kogutud andmetest luua vähese inimsekkumisega veekogu sügavuskaart ja seda kaardirakenduses kuvada. Samuti annab töö esimese eelhinnangu antud ärisuuna kõlblikkusest.

1.4. Autori roll

LAINE projekti meeskond töötab koos ettevõttes AS Datel ja omab praktilist kogemust tarkvaraarenduses. Meeskonnaliikmete igapäevatöö on lahenduste projekteerimine, programmeerimine ja testimine. Meeskonda kuulub kolm inimest ja nende rollid on järgnevad:

- Siima Tiitus – testija, konsultant;
- Veikko Danilas – analüütik, programmeerija, devops;
- Andreas Porman – arhitekt, programmeerija, tooteomanik.

Töö autor on antud projektis arhitekti rollis, kes koostab kogu rakenduse arhitektuuri, mis kirjeldab infosüsteemi ülesehitust ja aitab anda ülevaate tarkvara moodulite omavahelistest seostest ning jälgib (juhul, kui on kaasprogrammeerijaid), et peetakse kinni kokku lepitud programmeerimismustritest ja -tavatest. Programmeerijana on autori roll antud rakenduse serveri- ja klientrakenduse poole programmeerimine ning GIS-komponentide ja –töövoogude integreerimine ja loomine. Tooteomanikuna on töö autor vastutav toote väärtuse maksimeerimise ja arendustiimi töö eest.

Projekti meeskonna liikmete rollid, sh autori roll, on täpsemalt välja toodud tööde hierarhilise struktuuri (WBS) joonisel, mis on esitatud Lisa 2 juures.

1.5. Töö struktuur

Töö esimeses peatükis kirjeldatakse probleemi ja püstitatud ülesannet ning antakse ülevaade valdkonnast, töö skoobist ja autori rollist.

Metoodika ja analüüsi osas annab autor lühikese ülevaate milliseid arendusmetoodikaid projekti tootearenduse ja realiseerimise protsessis kasutati.

Kolmandas peatükis tuuakse välja AS Dateli koostatud äriülesanne ja antakse ülevaade juba turul olevatest sarnastest toodetest.

Disaini protsessi peatükis kirjeldatakse tootearenduse sisulist tööd ning tuuakse välja ja põhjendatakse rakenduse realiseerimiseks kasutatavaid tehnoloogiaid.

Prototüübi kirjelduse peatükis tutvustakse loodud prototüüplahendust, selle osi ning milliseid ärinõudeid suudeti ja millised ei suudetud rahuldada ja miks.

Kokkuvõttes käsitletakse töö tulemust ning vastatakse küsimusele kas töö eesmärk saavutati ja millist väärtust loodud prototüüplahendus ettevõttele pakub.

2. Metoodika ja analüüs

Uue toote/teenuse loomise esimeses etapis, tootearenduse faasis, kasutati ideede genereerimiseks ja sõelumiseks ning kontseptsiooni väljatöötamiseks peatükis välja toodud meetodeid ja lähenemisviise.

2.1. Arendusmetoodika ja nõuete kogumise tehnikate valiku põhjendus

Tarkvara loomise protsessiks pakub Sommerville (2016) välja kolme peamist mudelit:

1. Kosemudel – projekti skoop ja arendatav tarkvara on pandud paika enne koodi kirjutamist ja koodi kirjutamise ajal enam muudatusi ei tehta.
2. Inkrementaalne arendus – projekti skoop ja arendatav tarkvara ei ole detailideni teada ja koodi kirjutamise ajal tehakse nõuetes arendusprotsessi käigus muudatusi.
3. Integreerimine ja konfigureerimine – on taaskasutusele suunatud tarkvara loomine, kus uus süsteem luuakse olemasolevatest komponentidest ja selle loomiseks võidakse kasutada nii kosemudelit kui ka inkrementaalset arendust.

Välja toodud kolmest mudelist kasutab autor LAINE projektis inkrementaalset arendust, sest prototüüp luuakse nullist ning kliendiga konsulteerides nõuded arendusprotsessi jooksul muutuvad.

Nõuete kogumise tehnikad valiti töö käigus jooksvalt ning eelkõige tuginedes meeskonnaliikme Siima Tiituse kogemusele ja soovitudele.

2.2. Valitud projekti arendusmetoodika

Projektijuhtimise metoodikaks valiti agiilne lähenemisviis, kuna juhindutakse iteratiivsetest ja inkrementaalsetest tarkvaraarenduse meetoditest. Ghahrari (2019) järgi käsitletakse iteratiivset ja inkrementaalset mõistet järgnevalt:

- Inkrementaalne tähendab, et iga järgmine tarkvaratoode on eelmisest veidi parem (võimekam, töökindlam, enamate võimalustega jne) ehk uut funktsionaalsust lisatakse tarkvara väikeste osade kaupa.

- Iteratiivne tähendab, et arendusprotsessi peamisi tegevusi (nagu näiteks nõuete kavandamine, disain, koodi kirjutamine, testimine) teostatakse korduvalt ja kui iga arendustsükkel lõpeb, on tarkvaratoode ka inkrementaalselt parendatud.

Iteratiivse tarkvara arenduse idee on süsteemi pidevas, kindlate tsüklite järgi ning väikeste osade kaupa arendamises. See on hea sellepärast, et see võimaldab arendajatel saada kiiret ja kohest tagasisidet arendatud süsteemi kohta ning sellele infole tuginedes täiustada arenduses olevat süsteemi. Agiilse lähenemise peamine idee on luua kliendile minimaalne töötav versioon. Selle eesmärk on koguda võimalikult varakult tagasisidet, mille põhjal saab loodud süsteemi muuta ja täiustada. Varajane kliendi tagasiside on vajalik, et vältida kuudepikkust tarkvara arendust, mille tulemuseks on süsteem, mis tegelikult ei vasta kliendi soovidele ja ootustele (Ghahrai, 2019).

Agiilset lähenemisviisi toetab eepik (epic), mis on mahukas kasutuslugu (Standardipõhine tarkvaratehnika sõnastik). Eepik on kasutuslugu, mille saab jagada eraldi väikesteks kasutuslugudeks (Cohn, 2004). Eepikud vastavad järgmisele struktuurile:

- *As a <type of user> / Kasutajana <kasutaja roll/tüüp>*
- *I want to <perform some task> / soovin ma <tegevus>*
- *So that I can <achive some goal> / et saaksin <saavutada tulemuse>*

2.2.1. MoSCoW

Lähteülesandes kirjeldatud tööde prioritseerimiseks kasutati MoSCoW meetodit. MoSCoW meetod on projektijuhtimise ning tarkvaraarenduse jaoks kasutatav prioritseerimise meetod, mida kasutatakse erinevate osapoolte puhul ühisele arusaamisele jõudmiseks nõuete tähtsustest (MoSCoW Prioritisation). Mõiste MoSCoW on akronüüm prioritseerimise kategooriatest (*Must have, Should have, Could have, Won't have*) ja tähendavad *Agile Business*'i õppematerjali järgi järgmist:

- *Must have* - nõuded on toote tarnimiseks kriitilise tähtsusega. Kui kasvõi üks *Must have* nõue on täitmata, ei ole toote tarnimine võimalik. *Must have* nõudeid on võimalik tõsta ümber madalamatesse kategooriatesse, eeldades, et on olemas nõusolek kõikidelt seotud osapooltelt, kui näiteks uued nõuded loetakse eelmistest tähtsamaks.
- *Should have* - nõuded on olulised, kuid neid pole vajalik nii kiiresti tarnida. Kuigi *Should have* nõuded võivad olla sama tähtsad kui *Must have*, ei ole need ajaliselt sama

tähtsad, või on nende täitmiseks mõni muu viis, seega saab neid edasi lükata edaspidise tarnejani.

- *Could have* - nõuded on soovitatavad, kuid mitte vajalikud, ning võivad parandada kasutajakogemust või klientide rahulolu vähese kuluga. Need lisatakse tavaliselt siis, kui on piisavalt aega ja ressursse.
- *Won't have* - nõuded on kokkulepitult vähem kriitilised või pole tol hetkel ajakohased. Selle tulemusena ei ole *Won't have* nõuded lisatud tulevase tarneaja kavasse. *Won't have* nõuded suure tõenäosusega jäetakse tootest kas üldse välja või lükatakse edasi, et nende hilisem kaasamine läbi rääkida.

2.2.2. WBS

WBS on tehnika, mis aitab jagada töö või projekti väiksemateks osadeks. WBS on kasulik ja tähtis projektijuhtimistehnika, kuna võimaldab lihtsamini hoomata eesseisvaid vajalikke töid. Väiksemateks ülesanneteks jagatud projektid võimaldavad töid paremini hinnata ja täitjate vahel jaotada (What is a Work Breakdown Structure?).

2.3. Nõuete kogumine

Nõuded jaotatakse enamasti kaheks:

- Funktsionaalsed nõuded – kirjeldavad tarkvara või selle osa käitumist.
- Mittefunktsionaalsed nõuded – kirjeldavad infosüsteemi tausta võimekusi (nt. turvalisus, kasutatavad tehnoloogiad, käideldavus).

Nõuete väljatöötamise käigus kogutakse informatsiooni loodava tarkvara ja olemasoleva (reaalelus toimiva) süsteemi kohta. Nõuete kogumisel tuleb kõik kogutud nõuded dokumenteerida ja süstematiseerida (Tarkvaraprotsess_tegevused_2018).

Tabelis 1 on välja toodud kvaliteedimudeli standard ISO/IEC 25010:201 nõuete kohta.

Tabel 1. Süsteemi / tarkvaratoote kvaliteet (EVS-ISO/IEC 25010:2011põhjal)(Tepandi, 2020).

Kategooria	Alamkategooria
Funktsionaalne sobivus	<ul style="list-style-type: none"> • Funktsionaalne täielikkus • Funktsionaalne õigsus • Funktsionaalne kohasus
Soorituse tõhusus	<ul style="list-style-type: none"> • Ajaline käitumine • Ressursikasutus • Suutvus
Töökindlus	<ul style="list-style-type: none"> • Küpsus • Tõrketaluvus • Taastuvus • Töökindluse vastavus
Ühilduvus	<ul style="list-style-type: none"> • Koosoluvõime • Koostalitlusvõime
Kasutatavus	<ul style="list-style-type: none"> • Kohasuse äratuntavus • Õpitavus • Käsitsetavus • Eksituskindlus • Kasutajaliidese esteetika • Hõlpsus
Turvalisus	<ul style="list-style-type: none"> • Konfidentsiaalsus • Terviklus • Salgamatus • Jälitatavus • Autentsus
Hooldatavus	<ul style="list-style-type: none"> • Modulaarsus • Taaskasutatavus • Analüüsitavus • Modifitseeritavus • Testitavus
Porditavus	<ul style="list-style-type: none"> • Sobitatavus • Installeeritavus • Asendatavus

2.3.1. Lean Canvas

Lean Canvas on ärimudeli lõuend, mis on kohandatud alustavatele ettevõtetele nende elutsükli varases staadiumis. See annab asutajatele esimese võimaluse dokumenteerida oma tulevase idufirma ärimudel, mis seejärel pidevalt areneb. See on kasulik kõigile, kes on seotud uue toote või teenuse turule toomisega, s.t ettevõtte omanikele, potentsiaalsetele investoritele ja partneritele (Riphyak, 2017).

2.3.2. Requirements Discovery Canvas

Requirements Discovery Canvas on visuaalne tööriist, mis aitab meeskondadel tarkvara nõudeid avastada ja korraldada. Ärimudeli lõuendist inspireerituna pakub see koostööraamistiku, mis viib ühise mõistmiseni, mida saavad kasutada nii vilgas kui ka traditsiooniline tarkvaraarenduse meeskond (Robinson, 2015).

2.3.3. MVP

Minimaalne töötav toode (*minimum viable product*) on esimene versioon tootest, mida klient saab kasutada ja millelt saab tagasiside kogumiseks piisavalt infot, et tootearendusega jätkata (Tammemets). Infosüsteemide analüüsi ja projekteerimise aine materjalidele tuginedes peab MVP idee faasis vastama järgmistele küsimustele:

1. Kes on sihtgrupp?
2. Mis on probleem?
3. Mis toob väärtust?
4. Mida saame selle abil teada?
5. Kas poolik lahendus on vastuvõetav?

2.3.4. Liftikõne

Liftikõne (*elevator pitch*) on enese, ettevõtte, teenuse, toote või äriidee hästi läbi mõeldud, sisutihe ja korduvalt harjutatud lühike tutvustus (Tammemets). Liftikõne koostamise mall (The Crossing The Chasm Elevator Pitch Template):

- *For* / Kellele
- *Who* / Kes
- *The what* / Mis
- *Is* / On
- *That* / Mis
- *Unlike* / Mitte nagu
- *The product/service* / Toode/teenus

3. Olemasolevate lahenduste analüüs

3.1. Äriülesande püstitus

Loodav lahendus LAINE pakub tootjaülest platvormi, et valmistada veekogu põhjast sonariga kogutud andmetest kliendile huvipakkuvat infot, mille põhjal on võimalik vastu võtta ärilisi otsuseid. Olgu infoks siis sügavusandmed, põhja kõlviku/põhja kõvaduse kaart, objektid veekogu põhjas või 3D mudel veekogust.

AS Dateli poolt püstitatud äriülesanne:

1. Kliendil on võimalik määrata ulatust, milles ta veekogu kaardistust soovib.
2. Me suudame orienteeruvalt hinnata tehnilise platvormi poolt toorandmete kogumiseks ning lõppkasutajale soovitud andmete tootmiseks vajalikku aega.
3. Me oleme võimelised looma aluskaardi vähemalt järgmiste andmetega:
 - 3.1. Sügavusandmed;
 - 3.2. Põhja pinnakatete eristus (n.ö veealune kõlvikute kaart);
 - 3.3. Erihuviobjektide võimalik tuvastamine (näiteks suuremad kivid, vrakid, metallobjectid, veealused tehisrajatised);
 - 3.4. Võimalusel põhja profiili 3D pilti genereerimine.
4. Meil on rakendus, mis võimaldab kogutud toorandmetest automaatselt või vähese inimsekkumisega toota punktis 3 loetletud veepõhja kaardi.
5. Meil on rakendus, mis võimaldab toodetud aluskaardi andmeid sirvida.
6. Sirvimisrakendus võimaldab veetaseme muutuseid emuleerida.
7. Sirvimisrakenduses on võimalus sisse laadida ka sõltumatuid 3D mudeleid (kas varasemaid mõõtmisi või näiteks projekteeritud tammi põhja) ning teostada nende võrdlust (näiteks arvutada kubatuuri muutuseid).
8. Meie andmekogumise ja –tootmise platvorm on arendatud sõltumatult konkreetsest füüsilisest liikurplatvormist, kuid testitud vähemalt ühel liikuvplatvormil.
9. On loodud demo kasutatavast platvormist (näiteks Rummu Karjääri põhjakaardi näitel osas, kus on veealune hoonestus).
10. Meil on ülevaade, mis on konkureerivate platvormide teenuse osutamise aeg, hind ning nende pakutava andmete detailsus (millised kategooriad) ja kvaliteet (lahutus/ täpsus).

Kuna tegemist on madala TRL taseme (TRL 5) arendusega, siis antud töö arvestab ISO/IEC 25010:2011 (ISO/EIC 25010:2011) nõuete kategooriast ainult funktsionaalse sobivusega ja selle alamkategooriatega (funktsionaalne kohasus, funktsionaalne täielikkus ja funktsionaalne õigsus).

TRL madala taseme tõttu käsitleb käesolev töö ITIL-i tavade ja protsesside (vt kogu ITIL joonist koos antud skoobis kasutatud osadega Lisas 3) juures teenuse kavandamise ja realiseerimise osa. Kavandamise protsessi osas täpsemalt disaini koordineerimist ning realiseerimise protsessis nõuete ja arhitektuuri haldamist.

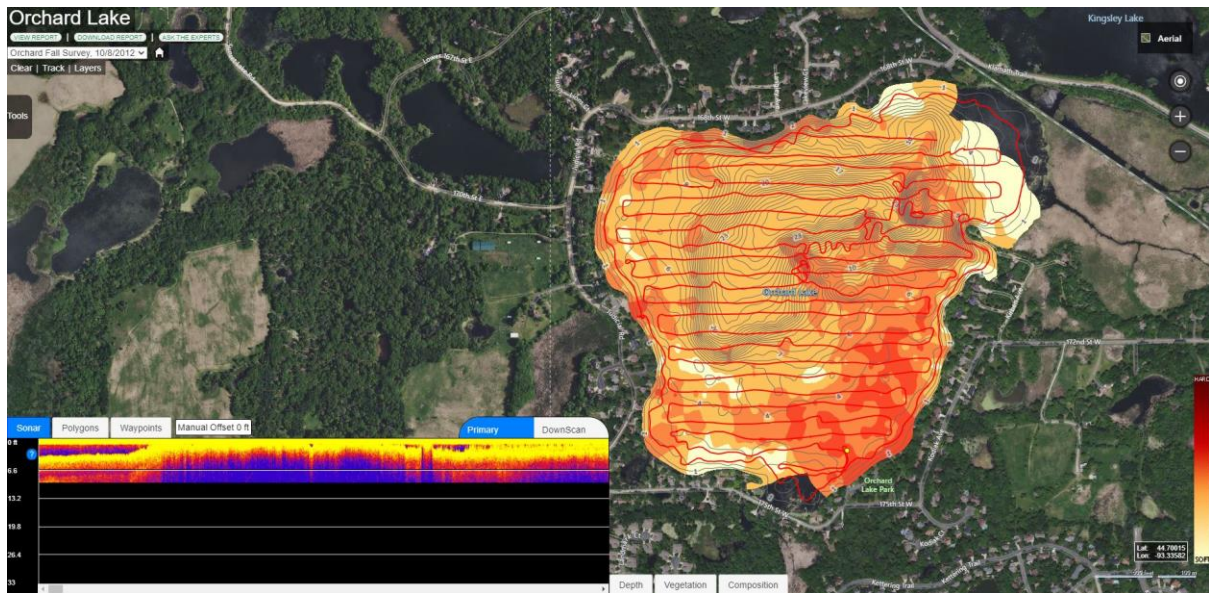
Nõuete prioritseerimine toimus MoSCoW meetodil ja selle tegevuse täpsem kirjeldus on leitav peatükis 4.1.

3.2. Turul olevate teenuste/toodete analüüs

Turul ringi vaadates leidis töö autor kaks sarnast toodet/teenust – Bio Base Maps by C-Map ja Garmin/Navioncsi ActiveCaptain. Antud tooted/teenused on seotud ainult kindlate tootjatega.

3.2.1. Bio Base Maps

Ettevõtte C-Map toode Bio Base Maps pakub ainult Lowrance või Simrad sonarite kogutud andmete visualiseerimise teenust kaardile. Andmetest luuakse sügavus-, põhja kõvadus- ja biomassi kaart. Samuti on võimalik kuvada läbitud teekonda ja näidata teekonna iga punkti kohta sonari logi pilti. Kasutajatel on võimalik (tasulise konto alusel) oma sonari logi (kogutud toorandmed) rakendusse üles laadida, kus neist moodustatakse eelnimetatud kaardikihid. Samuti on kasutajatel võimalus andmeid alla laadida ruumiandmete ja teksti failidena.

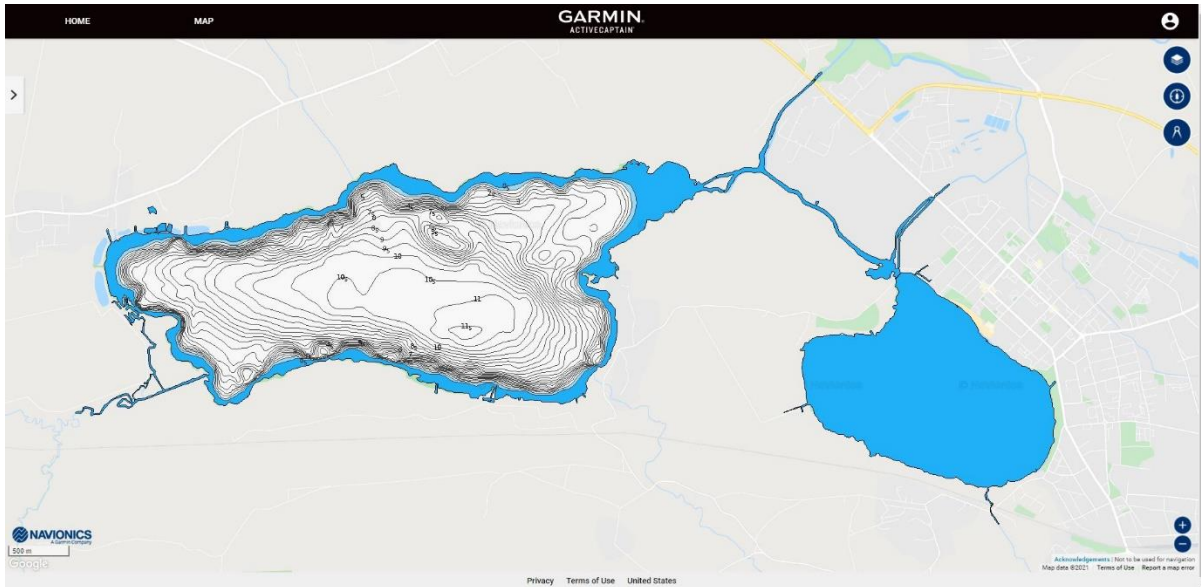


Joonis 1. Bio Base Map (kuvatõmmis Bio Base Maps rakendusest)

Bio Base Mapi omanikfirma C-Map omab ka toodet Social Map, kus kõiki Bio Base Mapi loodud kaardikihte kuvatakse. Antud kaarti on võimalik sirvida tasuta (näeb sügavusandmetega kaarte). Kaardi näide on esitatud Lisa 10 juures, põhja kõvadusinformatsiooni sisaldava kaardikihi nägemiseks tuleb luua eraldi tasuline konto.

3.2.2. Garmin ActiveCaptain

Garmini veebikaardirakendus, mis on eelkõige mõeldud veekogude navigatsiooniinfo jagamiseks, kuid kaardil on võimalik kuvada ka teatud veekogude sügavusandmeid. ActiveCaptain kogukonna liikmed saavad oma Garmin sonaritega kogutud andmeid rakendusse üles laadida ja need on näha kõigile kaardirakenduse kasutajatele. Garmini omandatud ettevõtte Navionics lisab võimekuse laadida rakendusse andmeid ka Raymarine, Lowrance, Simrad ja B&G sonari tootjate poolt.



Joonis 2. ActiveCaptain (kuvatõmmis ActiveCaptain rakendusest)

3.2.3. R-Süsteemid RAN, AEGIR ja HIS

R-Süsteemide loodud hüdrograafilise mõõdistuse tarkvara pakett, kuhu kuulub kolm eraldi seisvat rakendust – RAN, AEGIR ja HIS. RAN tarkvara peamine ülesanne on koguda ja salvestada batümeetria andmeid. Samuti võimaldab see mõõdistustöid planeerida ja kogutud andmeid visualiseerida. AEGIR tarkvara on mõeldud kogutud andmete kvaliteetseks järeltöötamiseks ja põhjaprofiili analüüsiks. AEGIR võimaldab tuvastada veealuseid objekte. HIS tarkvara ülesanne on kogutud ja salvestatud andmete haldamine ja kuvamine kaardil. RAN, AEGIR ja HIS ei ole vabavara. Eelnimetatud tarkvarade kasutaja on Transpordiameti merenduse ja veeteede teenistus.

3.3. Konkureerivate toodete võrdlus

Tabelis nr 2 on välja toodud konkureerivate toodete peamised näitajad: teenuse osutamise aeg, hind ning nende pakutavate andmete detailsus (kategooriad) ja kvaliteet (lahutus/täpsus). Kuna R-Süsteemide rakendustele töö autoril ligipääs puudus, siis neid tabelis nr 2 võrdluses välja ei toodud (puudus võimalus hinnata näitajaid).

Tabel 2. Konkureerivate toodete võrdlus

	Aeg	Hind	Detailsus	Kvaliteet
Bio Base Maps	6 tundi*	330 – 2321 EUR	Võimaldab kogutud toorandmetest luua veepõhja kaarti. Kliendil on võimalik määrata ulatust, milles ta veekogu kaardistust soovib. Sügavusandmed. Põhja pinnakatete eristus. Rakendus võimaldab toodetud aluskaardi andmeid sirvida.	Sügavus 1 m täpsusega.
ActiveCaptain	-	Tasuta**	Sügavusandmed***	Sügavus 0.5 m täpsusega. Kaardil navigatsiooni märgid.

* ligikaudne aeg, mis kulus veebirakendusse sonari logi faili (Lauga järve näitel) laadimisest kuni teateni, et andmetest on aluskaardid moodustatud

** veebikaardi rakenduses andmete sirvimine

*** mõeldud veekogude navigatsiooniinfo jagamiseks, kaardil on võimalik kuvada ka teatud veekogude sügavusandmeid

Bio Base Maps võimaldab ärinõuetes täita mitut tähtsat punkti, kuid ei suuda toota 3D pilti ja ka kaartide tootmine on seotud ainult kindla tootja sonarandmetega. Samuti ei võimalda Bio Base Maps rakendusse laadida sõltumatuid kõrgusmudeleid ja neid omavahel võrrelda ning ei saa simuleerida erinevaid veetasemeid. Garmin ActiveCaptain kuvab ainult veekogu sügavusandmeid ja on samamoodi seotud kindlate tootjatega. Seetõttu otsustati luua rakendus, millega saab rahuldada kõiki ettevõtte AS Datel poolt esitatud ärinõudeid. Täpsem võrdlustabel on välja toodud Lisas 8.

4. Disaini protsessi kirjeldus

Projekti alguses koostas töö autor koos meeskonnaliikme Siima Tiitusega esialgsed eepikud, *Lean Canvas*'e, *Requirements Discovery Canvas*'e, MVP ja liftikõne. Eelnimetatud materjale kasutas lähtematerjalina projekti meeskonnaliige Veikko Danilas projektile analüüsi koostamiseks ja oma magsitritöö „Mehitamata seadmete vahendusel teostatava veekogude põhja seire ja kaardistuse võimekuse analüüs“ tarbeks.

4.1. Nõuete kogumise ja prioritseerimise tööprotsess

LAINE projekti ülesande püstitusega alustati detailsemalt septembris 2019. Töö toimus projekti meeskonnaliikmete vahel nii individuaalselt kui töötoa-vormis koosolekutena. Koosolekuid peeti nii füüsiliselt ühes ruumis viibides kui ka kasutades suhtlusplatvorme Skype ja MS Teams. Esialgu toimusid koosolekud ebaregulaarselt, vastavalt vajadusele. Tavaliselt määrati käimasoleval koosolekul järgmise koosoleku aeg. Ülesande ja tööde täpsustudes hakkasid koosolekud toimuma regulaarselt kord nädalas.

Esialgsete nõuete kogumine toimus suures osas meeskonnatööna töötoalaadsetel koosolekutel. Kõigepealt arutati rakenduse põhikontseptsiooni üle ning koostati/kirjeldati äriülesanne. Samuti loodi esimesed eepikud, ideefaasi MVP ja liftikõne. Äriülesande täpsustamiseks mõeldi läbi valdkonnad, kus võiks antud toode/teenus huvi pakkuda. Otsustati, et parima tagasiside äriülesande kirjeldamiseks saab, kui saata loodava rakenduse idee potentsiaalsetele klientidele. Ideest huvitatud klientidega korraldati kohtumised, kus tutvustati uuesti omapoolset ideed ning kuulati klientide soove. Koos arutati, mida on mõistlik ja mida mitte antud teenusega teha. Toimunud kohtumised protokolliti ja protokollid korrastati. Kõigi tagasisidet andnud võimalike klientidega kohtumiste järel valiti välja üks klient, kelle ülesannet hakati lahendama. Selle kliendiga jõuti täpsustada ülesannet, täiendada ja muuta esmaseid eepikuid, täita *Lean Canvas* (vt Lisa 5), *Requirements Discovery Canvas* (vt Lisa 5) ja kohandada tootearenduse idee faasi MVP ning korraldada mitu kohtumist enne, kui juriidilistel põhjustel koostöö katkes. Seetõttu korraldati kogu protsessi võimalike klientide ringi laiendades ning jõuti taaskord ühe potentsiaalse kliendi ja tema ülesande juurde. Paraku seegi koostöö katkes, sest ülesande täpsustudes jõuti arusaamale, et antud lahendus ei pakuks ei kliendile ega teenuse loojale midagi uut. Lõpuks otsustati, et kliendiks jääb ettevõtte AS Datel ise. Tootearendusfaasis loodi Datelile idee tutvustamiseks ning prototüübi aluspõhja loomiseks ja programmeerimiseks idee MVP, liftikõne ja esialgsed eepikud (vt Lisa 6).

AS Datelile loodud MVP idee:

1. **Sihtgrupp:** Isikud, kellel on vaja langetada strateegilisi ja ärilisi otsuseid vee all asuvate objektide andmete põhjal.
2. **Probleem:** Info saamine on vaevaline, kuna tuleb kasutada mitut tehnoloogiat eraldi ja vastus käsitsi kokku panna.
3. **Mis toob väärtust?** Platvorm, kuhu on eraldi seisvad tehnoloogiad ühtseks tööriistaks integreeritud ja mis võimaldab langetada otsuseid kogutud andmete põhjal.
4. **Mida saame selle abil teada?** Saame vastuse küsimustele: 1) kas keegi soovib loodavat toodet kasutada; 2) kas loodav toode katab kliendi vajadused.
5. **Kas poolik lahendus on vastuvõetav?** Jah, kui platvorm võimaldab mugavalt eritehnoloogiad kasutades huvipakkuvaid andmeid koguda.

AS Datelile loodud liftikõne:

FOR kõigile,

WHO on vaja koguda vee alt andmeid otsuste tegemiseks,

LAINE

IS A platvorm

THAT võimaldab automaatset ja autonoomset andmete kogumist ja kaardistust,

UNLIKE praegused lahendused,

THE PRODUCT/SERVICE kiirendab ja mitmekesistab otsinguid, aitab andmeid turvaliselt ning mugavalt koguda ja kaardistada ühes rakenduses.

Esitatud liftikõne kasutati LAINE idee mahamüümiseks ettevõttele AS Datel. Töö käigus täpsustus projekti skoop ja liftikõne osad. Eelkõige muutus:

- Kellele/*For* – AS Dateli kaugseire kliendid;
- Mis/*That* – veekogude põhja kaardistust;
- Toode/teenus/*Product/Service* – on sõltumatu kindlast tootjast ja füüsilisest liikurplatvormist.

Dateli koostatud lähteülesande prioritiseerimine toimus meeskonna ühise koosoleku käigus, kus hakati punkt-punktilt lähteülesande alamülesandeid määrama MoSCoW kategooriate alla, mis on esitatud Lisa 7 juures. Iga meeskonnaliige põhjendas oma arvamust vastava

alamülesande kohta ning koos otsustati, mis kategooriasse antud soov minimaalse töötava versiooni ehitamiseks kuulub.

4.2. Valitud tehnoloogiad

LAINE platvormi realiseerimiseks kasutatavad tehnoloogiad on valitud lähtudes sellest, et ettevõttes AS Datel kasutab töö autor antud tehnoloogiaid oma igapäevatöös ning kuna loodava töö tulemus on lähtepunkt edasistele arendustele, siis rakenduse realiseerimisel on ettevõtte meeskonnal olemas antud tehnoloogiatel kompetents. Samuti suudab töö autor nende tehnoloogiatega kõige kiiremini rajada lahenduse prototüübi.

Rakenduse *back-end* arendamiseks on kasutatud järgnevaid tehnoloogiaid:

- Java – kasutusel on Java versioon 15, mis rakenduse loomise hetkel oli Java viimane versioon. Java on kasutusel, kuna on platvormist sõltumatu objektorienteeritud programmeerimiskeel ja sobilik kasutada klient-server rakenduse serveripoolse keelena (Introduction to Java TM Technology). TIOBE 2021 märtsikuise indeksi alusel on Java maailmas populaarsuselt teine keel, seega on kindlam leida Javas programmeerimisel eettulevate probleemide lahendamiseks abi.
- *Spring* raamistik – kasutusel on versioon 2.4.3. *Spring* raamistiku põhifunktsioone saab kasutada ükskõik milline Java-rakendus ning raamistiku erinevad moodulid pakuvad erinevaid teenuseid (Spring Framework Documentation). Antud rakenduse prototüübi juures on kasutusel *Spring Web model-view-controller* (MVC) raamistik, mis võimaldab luua veebirakendusi ning toetab *RESTful* veebiteenuseid (Spring Framework Documentation). Samuti on kasutusel *Java Database Connectivity* (JDBC), mis võimaldab ligi pääseda andmebaasis asuvatele andmetele (Spring Framework Documentation).
- Spring Data JPA – on osa *Spring* raamistikust ning selle abil on võimalik konverteerida andmebaasi olemimudel Java objektideks ja vastupidi, luues võimaluse lugeda ja kirjutada andmeid andmebaasi (Gierke et al, 2021).

Rakendus *front-end* arendamiseks on kasutatud järgnevaid tehnoloogiaid:

- React - kasutusel versioon 17. React (ka React.js või ReactJS) on JavaScript raamistik veebirakenduste kasutajaliideste loomiseks (Ashwini, 2017). Loodav rakendus on

osaliselt *model-view-controller*-tüüpi ning React vastab selles *View* osale (Ashwini, 2017). Samuti on React populaarne (Daityari, 2021), mis võimaldab leida suurt hulka abistavaid materjale ette tulevate probleemide tekkimisel.

- React Semantic UI - kasutusel versioon 2.0.3 on veebirakenduste kujundamise raamistik, mis põhineb Reactil (Semantic UI React).
- Openlayers - kasutusel versioon 6.0.5, on JavaScriptil põhinev vabavaraline teek, mille abil on võimalik kuvada ruumiandmeid (kaardipõhistel) veebirakendustel (Openlayers). Openlayers kuulub OSGeo sihtasutuse alla ning on valitud, kuna rakenduses on kasutusel veel teisigi OSGeo tooteid, mis muudab nende omavahelise integreerimise lihtsamaks.
- webpack – kasutusel on versioon 5, on JavaScripti moodulite pakkija, mis aitab muuta JavaScripti rakenduse modulaarseks (Concepts). Webpack sisaldab ka veebiserverit, mis tarkvara arendusprotsessi käigus jooksutab klientrakendust (Concepts).

Andmebaas:

- PostgreSQL - kasutusel versioon 13, on vabavaraline relatsioonilise andmebaasi haldamise süsteem (What is PostgreSQL?).
- PostGIS – kuulub OSGeo tooteperekonda ja tagab/annab PostgreSQL andmebaasile toe salvestada, säilitada ja töödelda ruumiandmeid (PostGIS).

GRASS GIS - kasutusel versioon 7.6.1, on vabavaraline GIS-tarkvara, mida kasutatakse ruumiandmete haldamiseks, töötlemiseks, analüüsimiseks ja kuvamiseks. GRASS võimaldab luua andmetöötlemise skripte ja neid integreerida erinevate rakendustega. GRASS kuulub samuti OSGeo tooteperekonda (GRASS).

Geoserver - on vabavaraline Javaal põhinev serverrakendus, mis võimaldab jagada, käsitleda ja muuta ruumiandmeid. Kuulub OSGeo tooteperekonda (Geoserver).

Tööde jaotamiseks, lähtekoodi halduseks, dokumentatsiooniks ja projektijuhtimiseks kasutatakse Atlassian tooteperekonnast järgnevaid tooteid:

- Jira – tööde planeerimine, jaotamine, jälgimine.
- Confluence – projekti dokumentatsioon.
- Bitbucket – lähtekoodi haldus.

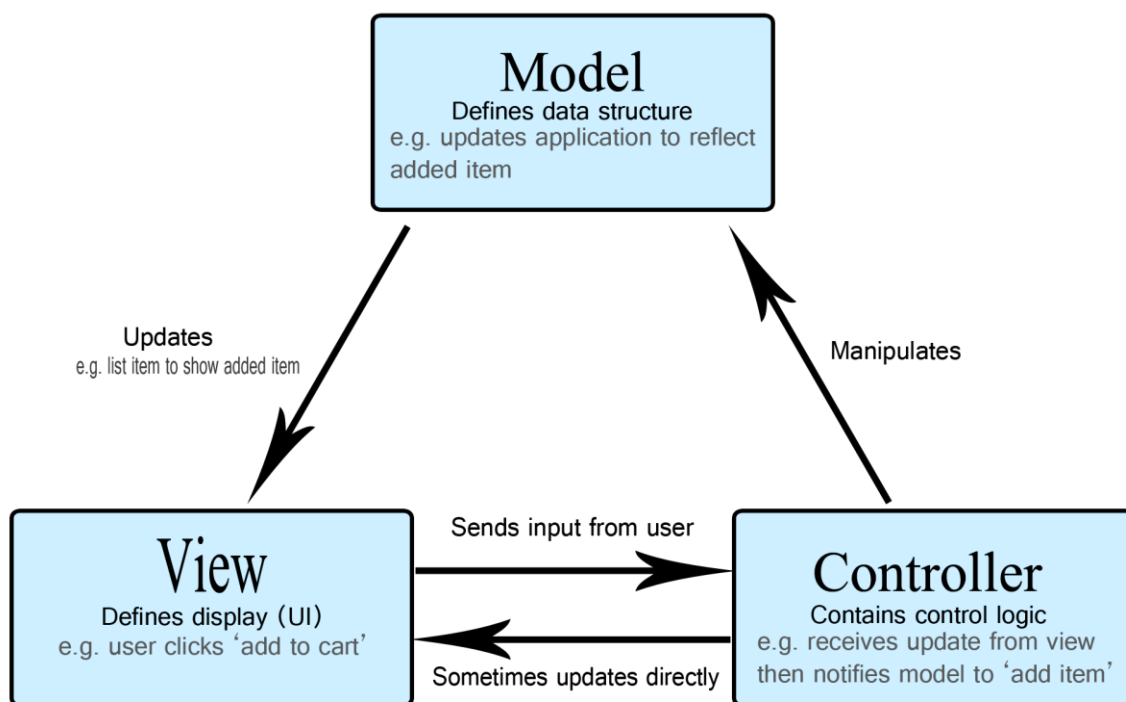
4.3. Valitud tarkvara arenduse mustrid

Prototüübi rakenduse loomisel on kasutatud MVC, *RESTful* ja *Domain Modal* mustreid.

MVC – on tarkvara arenduse muster, mida tavaliselt kasutatakse kasutajaliideste, andmete ja kontrollloogika rakendamiseks. See rõhutab tarkvara äri loogika ja kuva eraldamist, mis tagab parema tööjaotuse ja parema hoolduse (MVC).

MVC mustri kolm osa:

1. *Model* – selles osas hallatakse andmeid ja toimub äri loogika.
2. *View* – selles osas kuvatakse andmeid ja rakenduse kasutajaliidest.
3. *Controller* – ühenduslüli mudeli ja vaate vahel.



Joonis 3. MVC diagramm (MDN Web Docs)

Mudel määratleb, milliseid andmeid rakendus peaks sisaldama. Kui nende andmete olek muutub, teavitab mudel tavaliselt vaadet (nii, et kuva võib vastavalt andmetele muutuda) või mõnikord ka kontrolleri (kui uuendatud vaate juhtimiseks on vaja teistsugust loogikat) (MVC).

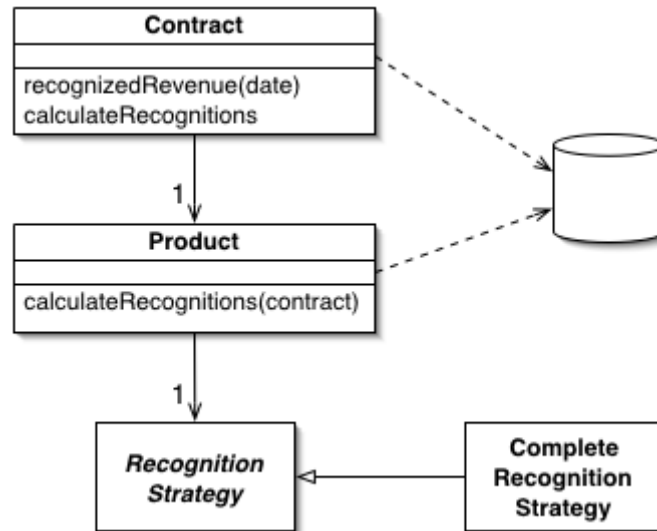
Vaade määratleb, kuidas rakenduse andmeid kuvatakse (MVC).

Kontroller sisaldab loogikat, mis värskendab mudelit või vaadet vastusena rakenduse kasutajate sisendile (MVC).

REST – tarkvaraarhitektuuri laad, mis viitab tarkvaraarhitektuuri disaini nõuetele, mis võimaldavad luua tõhusaid, usaldusväärseid ja skaleeritavaid hajussüsteeme (MVC). Roy Fieldingu järgi on REST tarkvaraarhitektuuril 6 nõuet, mis peavad olema täidetud:

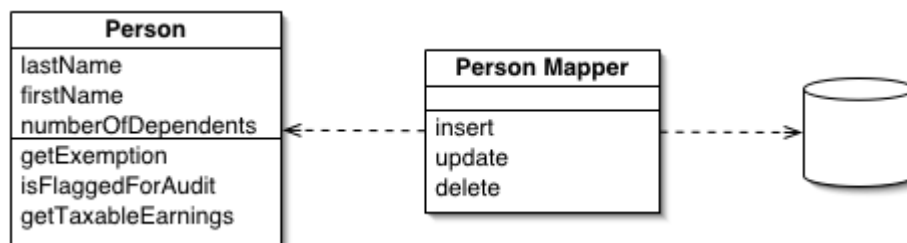
1. *Client-Server*: kasutajaliides peab olema andmete salvestamisest eraldatud.
2. *Stateless*: iga päring peab sisaldama kogu päringu mõistmiseks vajalikku teavet ega tohi kasutada serveris salvestatud konteksti.
3. *Cacheable*: vahemälupiirangud nõuavad, et päringu vastuses olevad andmed oleksid vahemällu salvestatavad või vahemällu mittekuuluvad. Kui päringu vastus on vahemällu salvestatud, antakse kliendi vahemälule õigus vastuse andmeid hilisemate samaväärsete taotluste jaoks uuesti kasutada.
4. *Uniform Interface*: REST komponentide vahel on ühtne liides:
 - 4.1. Iga ressursi päring on identifitseeritav URIga. Päritav info on kontseptuaalselt eraldatud algallikast ja saadetakse kliendi HML, XML või JSON kujul.
 - 4.2. Klient võib piisava info korral muuta andmeid serveris, kui tal selleks luba on.
 - 4.3. Iga saadetud teade omab endas piisavalt informatsiooni, et server või klient teab, kuidas vastavat teadet töödelda.
 - 4.4. Klient edastab teavet sõnumi keha, päringu parameetrite ja URI nime kaudu. Server edastab teavet sõnumi keha, vastuskoodide ja –päiste kaudu.
5. *Layered Style*: kihiline süsteemistiil võimaldab arhitektuuri koostada hierarhilistest kihtidest, piirates komponentide käitumist nii, et iga komponent ei saaks "näha" kaugemale otsesest kihist, millega nad suhtlevad.
6. *Code on demand*: REST võimaldab kliendi funktsionaalsust laiendada, laadides alla koodi ja käivitades selle väikeste äppidena või skriptide kujul. See lihtsustab ja vähendab kliendi eelrakendamiseks vajalike funktsioonide arvu.

Domain Model – on kontseptuaalne mudel, mis esindab andmebaasi olemeid ja nendevahelisi suhteid koodis objektidena, olgu see sama suur kui ettevõtte või sama väike kui üks rida tellimusvormil (Fowler, 2003).



Joonis 4. Domain Model (Fowler, 2003)

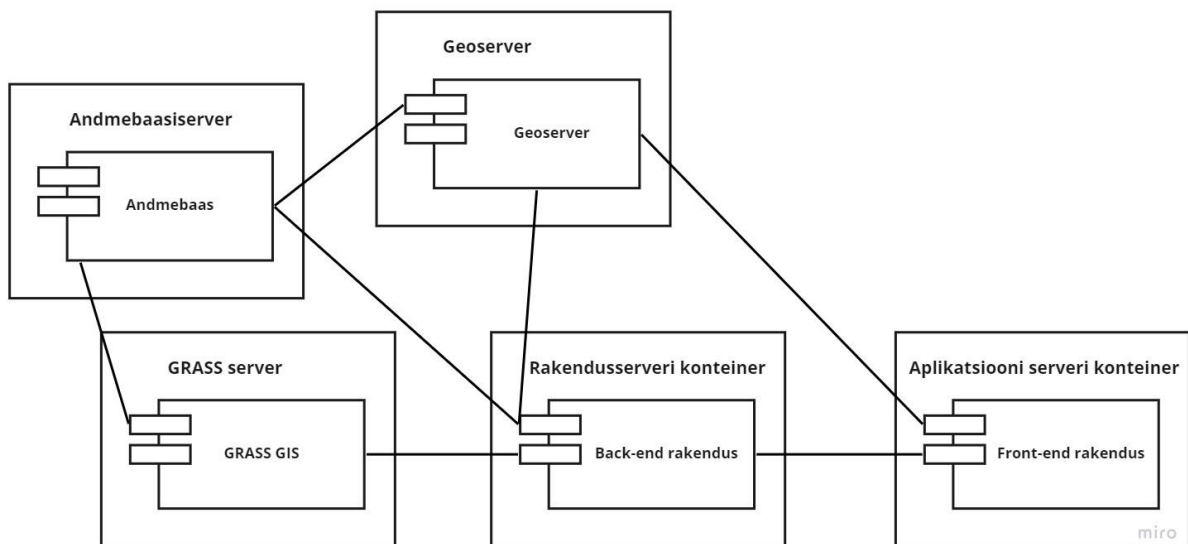
Fowler jagab domeeni mudelid kaheks: lihtne domeeni mudel, mis näeb samasugune välja nagu andmebaas (igale andmebaasi olemile vastab üks domeeni mudel); „rikas“ domeeni mudel võib välja näha väga erinev andmebaasist tänu objektorienteeritud programmeerimise paradigmat. „Rikka“ domeeni mudeli andmebaasi olemit vastu kujutamiseks on tarvis *Data Mapper* (arhitektuuri muster, mis tagab objekti andmebaasi kirjutamise, lugemise, muutmise ja kustutamise õigused ning „kaardistab“ objekti ja andmebaasi olemit õiged väljad).



Joonis 5. Data Mapper (Fowler, 2003)

4.4. Rakenduse arhitektuur

Rakenduse arhitektuur peab tagama programmi efektiivse töö, samas tagama ka selle turvalisuse ja kasutaja mugavuse. Arhitektuur kirjeldab rakenduse, serveri ja andmebaasi vahelist suhtlust (Banga, 2020). Paigaldusvaade on osa rakenduse arhitektuuridokumendist ja annab ülevaate süsteemi komponentide paiknemisest (virtualiseeritud) riistvaras.

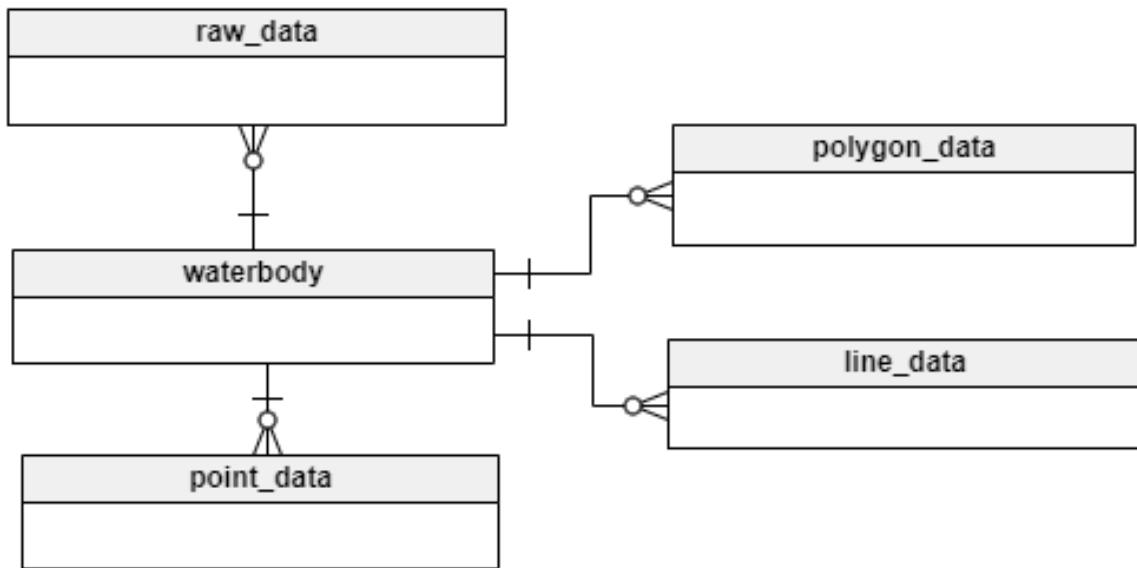


Joonis 6. Evičiuskeem

Meeskonnaliikmete vahel tööde jaotamiseks ja ettevõttele AS Dattel LAINE rakendusest ülevaate andmiseks loodi arhitektuuri kommunikatsiooniformaadis ülevaade (vt Lisa 9). Antud formaadi mall saadi Agu Leinfeldi käest ning see on segu aplikatsiooniarhitektuurist ja infoarhitektuurist. Aplikatsiooniarhitektuur kirjeldab, kuidas ettevõttes kasutuses olevad rakendused käituvad ning kuidas need omavahel ja kasutajatega suhtlevad. Samuti keskendub see rakenduste kasutatud ja toodetud andmetele, mitte nende sisemisele struktuurile (Applications architecture, 2021). Informatsiooniarhitektuur näitab, kuidas on süsteemis informatsioon grupeeritud, milline on süsteemi struktuur ja selles navigeerimise viisid (Information architecture, 2021).

4.5. Rakenduse andmemudeli kasutamine

Prototüübi jaoks koostati ka lihtne andmemudel. Joonisel 7 on loogilisel tasemel andmemudel ja atribuutidega täidetud variant on näidatud Lisa 11 juures. Andmemudel koostati selliselt, et olem *waterbody* kujutab endast keskset tabelit, milles hoitakse veekogu kohta andmeid ning olemites *polygon_data*, *line_data* ja *point_data* hoitakse ruumiandmeid vektorkujul. Ruumiandmete olemid on geomeetriatüübi järgi jagatud polügoon-, joon- ja punktandmete tabeliteks. Samuti loodi olem *raw_data*, kus hoitakse töödeldud toorandmeid. Olem *waterbody* on teiste olemitega seotud üks-mitmele seosega.



Joonis 7. ERD diagramm

5. Prototüübi kirjeldus

Prototüüplahendus koosneb klient-serverrakendusest, andmebaasist, geoserver ja GRASS GIS rakendusest. Andmekihtide ettevalmistamine toimub GRASS GIS nimelises rakenduses, kust valmiskihid laetakse otse Postgresql andmebaasi. Geoserver-rakendust kasutatakse ruumiandmete jagamiseks ja kuvamiseks WMS- ja WFS-teenustena. Serverrakendust kasutatakse andmebaasi ja geoserveriga suhtlemiseks ning klientrakendust andmete sisestamiseks, muutmiseks ja kuvamiseks.

5.1. Projekti püstipanek

LAINE projekti loomine algas AS Datel süsteemiadministraatoritelt Postgresql andmebaasi koos PostGIS võimekuse ja lähtekoodi repositooriumi ning sinna sisse kahe projekti asukoha tellimisega. Samal ajal loodi serverrakenduse põhi kasutades Spring initializer (spring initializer) keskkonda. Seal loodud Maveni projekt laeti alla ja imporditi Eclipse IDE-sse ning lisati git-i LAINE repositooriumisse projekti „laine“. Projektil on 2 põhilist haru: master ja develop. Neist esimest kasutatakse antud prototüüplahenduse juures lähtekoodi kokkuehitamise allikana ning teisest harust tekitavad arendajad alamharusid, mis oma töö valmides lisatakse tagasi põhiharusse. Klientrakenduse põhi loodi samamoodi nullist webpack-i projektina konsulteerides webpack-i kodulehe, internetis leiduvate erinevate blogide ning ettevõttes AS Datel varasemalt loodud webpack-i projektidega. Loodud rakenduse põhi lisati samamoodi git-i LAINE repositooriumisse projekti „*laine-web*“. GRASSi ja Geoserveri server olid ettevõttel AS Datel juba varasemalt üles seatud.

Kuna ruumiandmetega töötlemise skripte ei jõutud luua (täpsemalt punktis 5.3.) siis tuli GRASS GIS rakendus üles seada töö autori masinas. GRASSi ülsseadmine koosneb kolmest sammust, millest esimene on GRASSi andmebaasi loomine. Praktikas tähendab see lihtsalt arvuti failisüsteemis kaustale viitamine, kus GRASS hakkab hoidma kõiki andmeid, mida on tarvis, et oleks võimalik ruumiandmetega tööd teha. Järgmisena tuleb määrata projekti „asukoht“ (GRASSis *LOCATION*). *LOCATION* määrab ära käsitletava ala koordinaatsüsteemi, kaardi projektsiooni ja geograafilised piirid. LAINE projektis on selleks L-EST97 koordinaatsüsteem. Kolmandaks tuleb luua *MAPSET*, milles GRASS hoiab oma loogika alusel ruumiandmeid, mida kasutaja poolt loodud projektis kasutatakse.

Kõige aeganõudvam oli klientrakenduse tarbeks webpack-i konfigureerimine, kuna infot leidis internetis väga palju ja sealt vajaliku väljasorteerimine ning saadud juhiste järgimine ei toonud alati edu. Palju oli katse-eksituse meetodil tegemist. Üks probleem, mis tekkis webpack-i konfigureerimisel, oli minifitseeritud css-i failide lugemine React-Semantic UI tarbeks. Selleks otstarbeks mõeldud webpack-i „*style-loader*“ ja „*css-loader*“ ei hakanud korrektselt tööle ning probleem lahendati lõpuks „ajutiselt“ *index.html* faili päisesse React-Semantic UI teegi URL-i lisamisega.

5.2. Andmed

Prototüüplahenduse realiseerimiseks kasutas töö autor oma varasema Tartu Ülikooli Geograafia osakonna bakalaureusetöö „Kalaleidja abil veekogu sügavuskaardistamine Lauga järve näitel“ ning muude ülikooliga seotud projektide tarbeks kogutud sonarmõõdistuse andmeid. Andmed pärinevad vahemikust 2010-2012 ning on kogutud Lowrance HDS-5 WWB 50/200kHz kombineeritud kajalood/kaardiplotteriga. Lowrance-i sonar salvestab lisaks tervele hulgale andmetele asukoha koordinaadid, sügavusandmed ja tagasihajuvuse intensiivsuse, mis on vajalikud lähteandmed sügavus- ja kõlvikukaartide loomisel. Kahjuks sonari logi eksportimisel tekstifaili tagasihajuvuse intensiivsuse andmeid, mis on vajalikud veekogupõhja kõlvikukaardi loomiseks, aga ei salvestata (Herkül, 2014).

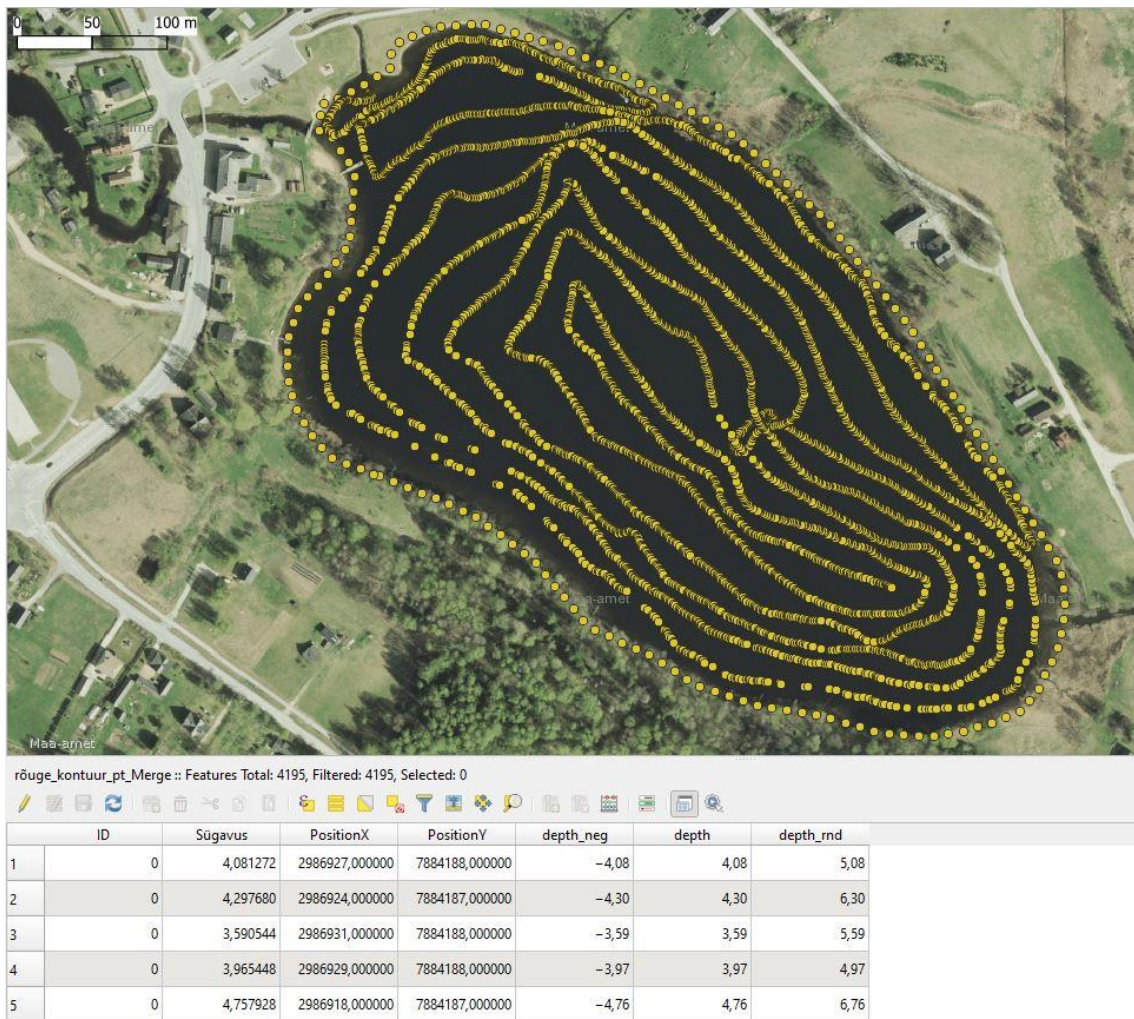
Toorandmete (vaata näidet Lisas 12) töötamise käigus jäetakse alles järgmised veerud: *Depth*, *PositionX*, *PositionY*, *SurfaceValid* ja *TopOfBottomValid*. Nendest kaks viimast kajastavad anduri ja veesamba sügavust. Nimetatud andmeväljade alusel on hea riknenud andmeridu eemaldada, sest vigaste andmeridade korral on antud väljade väärtus 0. Järgmise sammuna toimus andmete filtreerimine „*advanced filter*“ funktsiooni abil, kus loendi vahemikku peavad jääma X- ja Y-koordinaadid ning kriteeriumiks kogu tabeli neli veergu, samuti teha linnuke lahtrisse „unique records only“, saadakse tabel, kus osad read on puudu. Edasi tuleb saadud tabel kopeerida ja asetada uuele töölehele, funktsioonis „*paste special*“ teha linnuke „*values*“ lahtris. (Tõnisson, 2010.) Kuna sügavused olid jalgades, tuli need ümber arvutada meetermõõdustikku (1 jalg = 0,3048 meetrit) ning vähendada komakohti kahele. Kirjeldatud andmetöötamise töövoog programmeeriti ka prototüübi serverrakendusse, mis lisaks eelmainitule salvestab andmed andmebaasi.

5.3. GIS-lahendused

Koostatud lähteülesandes on mitu GIS-i alast ärinõuet, mida prototüüp peab suutma rahuldada. Töö autori käsutuses olevatele andmetele tuginedes ei saa aga kahjuks kõiki nõudeid täita:

1. Põhja pinnakatete eristust (n.ö veealune kõlvikute kaart) ei saa luua, kuna puuduvad tagasihajuvuse intensiivsuse andmed.
2. Erihuviobjektide võimalik tuvastamine (näiteks suuremad kivid, vrakid, metalloobjektid, veealused tehisrajatised). Antud ülesande tarbeks on tarvis külgvaatesonarit (Kaeser et al, 2013). Kaeser ja Litts toovad oma korraldatud koolituse „*An Illustrated Guide to Low-cost, Side Scan Sonar Habitat Mapping*“ materjalides välja töövoogu, kuidas külgvaatesonari andmetele tuginedes luua nii veealust pinnakatte erisust kui tuvastada erinevaid vee all asuvaid objekte.

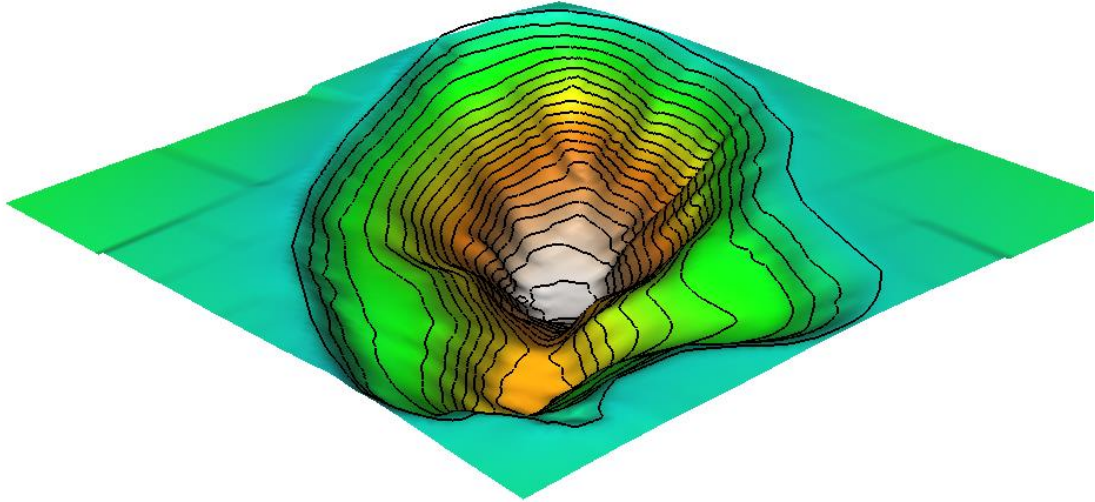
Loodud prototüüp prakenduse esmane nõue on võimekus luua veekogu põhjast erinevaid kaardikihte. Sügavuskaardi loomiseks on vaja veekogus mõõdetud sügavuspunkte. Punktis 4.2. loodud andmed loeti QGIS-nimelisse rakendusse ning nihutati idanihke ja põhjanihke abil paika ning viidi seejärel üle Eesti põhikaardi projektsiooni L-EST97 (Roosaare et al, 2019). Seejärel moodustati kaldajoonest 0-sügavusel asuvad punktid, mis konverteeriti joonest tihedusega 1 punkt iga 10 meetri kohta. Nii kaldajoone punktid kui ka sügavuspunktid liideti ühtseks tabeliks.



Joonis 8. Esri shape fail koos atribuutandmete tabeliga

Edasi salvestati loodud kiht Esri *shape* failiks ja imporditi GRASS GIS rakendusse. GRASS-is moodustati „*v.surf.rst*“ funktsiooniga sügavuspunktidest rasterkiht, millest omakorda moodustati „*r.contour*“ funktsiooniga veekogu sügavuskontuurjooned. Veekogu sügavuspindade moodustamiseks tuli kontuurjoonte joonandmed konverteerida polügonideks ehk pindadeks. Need joon- ja polügonandmed salvestati andmebaasi ning tehti geoserver rakendusele nähtavaks, et loodud kihte saaks kasutada WMS- ja WFS-teenustes.

Saadud sügavuskaartidest moodustati prototüüplahenduse tarbeks GRASS GIS 3D tööriistadega ruumilised pildid veekogu põhjast. Parema visuaali saamiseks kohandati „*Z-exag*“ väärtusi, mistõttu pildid ei ole mõõtkavaliselt (z-telg) täpsed (kuid on autori hinnangul ülevaate saamiseks piisavad).



Joonis 9. Rõuge järve põhi 3D

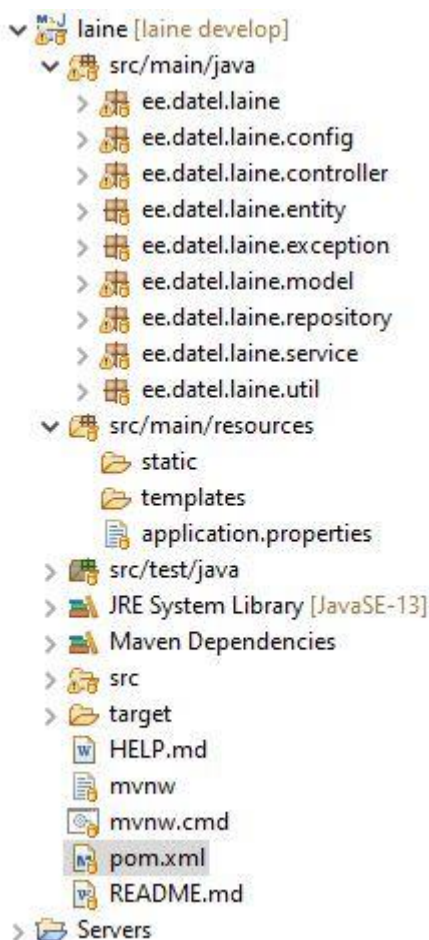
Veetasemete muutuste simuleerimiseks kasutati Maa-ameti geoportaalist kättesaadavat 10-meetrise piksliga DEM-mudelit ja Eesti põhikaardilt pärinevaid veekogu veetaseme nii-öelda „0-kõrguse“ andmeid. Neid lähteandmeid kasutades simuleeriti GRASS „*r.lake*“ funktsionaalsust kasutades Lauga järvel erinevaid veetasemeid, milleks olid 0.30, 0.60 ja 0.90 meetrit üle järve „0-kõrguse“. Saadud rasterkaardid viidi üle vektorandmete formaati polügoonideks ning salvestati andmebaasi.

Kõrgusmodelite võrdlemise ja kubatuuride muutuste arvutamise näitlikustamiseks tegi töö autor Rõuge Suurjärve sügavusandmetele tuginedes uue kaardikihi, mis simuleeris veekogu põhja kõrguse/sügavuse muutust. Saadud kiht imporditi GRASS GIS rakendusse ja „*v.surf.rst*“ funktsiooniga loodi sellest uus rasterkiht. Seejärel kasutati „*r.mapcalc*“ funktsionaalsust, et arvutada välja kahe pinna vahe. Antud vahest moodustati kontuurjooned, millest omakorda moodustati polügoonid. Igale polügoonile arvutati pindala ja ruumala. Saadud kaardikihid salvestati andmebaasi.

Probleemiks osutus toorandmetest sügavuskaardi automaatse protsessi loomine, kuna töö autori käsutuses olevatele andmetele tuginedes ei leitud ühtegi universaalset igale veekogu punkt pilvele sobivat töökäiku. Kui protsess sobiski ühtedele veekogu andmetele, siis teise veekogu andmetega tekkis kuskil mingi viga ja lõpptulemus oli vale. Mingis etapis vajab andmestik käsitsi lähenemist (näiteks oli tarvis käsitsi kontuure juurde lisada, ära kustutada või hoopis lisada mõne sügavuse polügoon), mistõttu esialgu toodetakse antud kihte käsitsi.

Kuna sügavuskaartide tootmise universaalse töövoogu peale kulus rohkem aega, kui arvestatud, ning automaatset protsessi ei suudetud luua, jäid ka teised GIS-töövood prototüübi lahenduses käsitööetappi. Neid tulemusi on siiski võimalik klientrakenduses kuvada. Sellega soovib/soovis töö autor näidata, et esitatud ärinõudeid on veekogu sügavusandmeid kasutades võimalik täita.

5.4. Serverrakendus



Joonis 10. LAINE projekti struktuur

Prototüüplahenduse serverrakendus võimaldab kasutajal suhelda andmebaasiga (andmeid lugeda, lisada, muuta, eemaldada). Samuti on serverrakenduse ülesanne käituda vahemehena klientrakenduse ja geoserveri teenuste vahel, et vahendada WMS- ja WFS-teenuseid. Serverrakendusse on programmeeritud kaasmeeeskonnaliikme Veikko Danilase peatükis 4.2. kirjeldatud andmetöötluse töövoog, mida on täiendatud võimekusega töödeldud andmed salvestada andmebaasi, kust neid saab kasutada GIS-töödeks.

Joonisel 10 on projekti struktuuris näha tarkvara arenduse MVC mustri osi *controller*-i ja *model*-i näol. *RESTful*-i näited on *controller*-is ja *service* paketi klassis *WfsService.java* näitel (Joonis 11).


```

public void wfsQuery(String type, HttpServletResponse response) throws
Exception {
    StringBuilder sb = new StringBuilder();

    sb.append(wfsUrl).append("&cql_filter=type=%27").append(type).append
("%27");

    String wfsFeaturesUrl = sb.toString();

    HttpPost request = new HttpPost(new
URIBuilder(wfsFeaturesUrl).build());
    request.setHeader("Content-Type", "application/json");

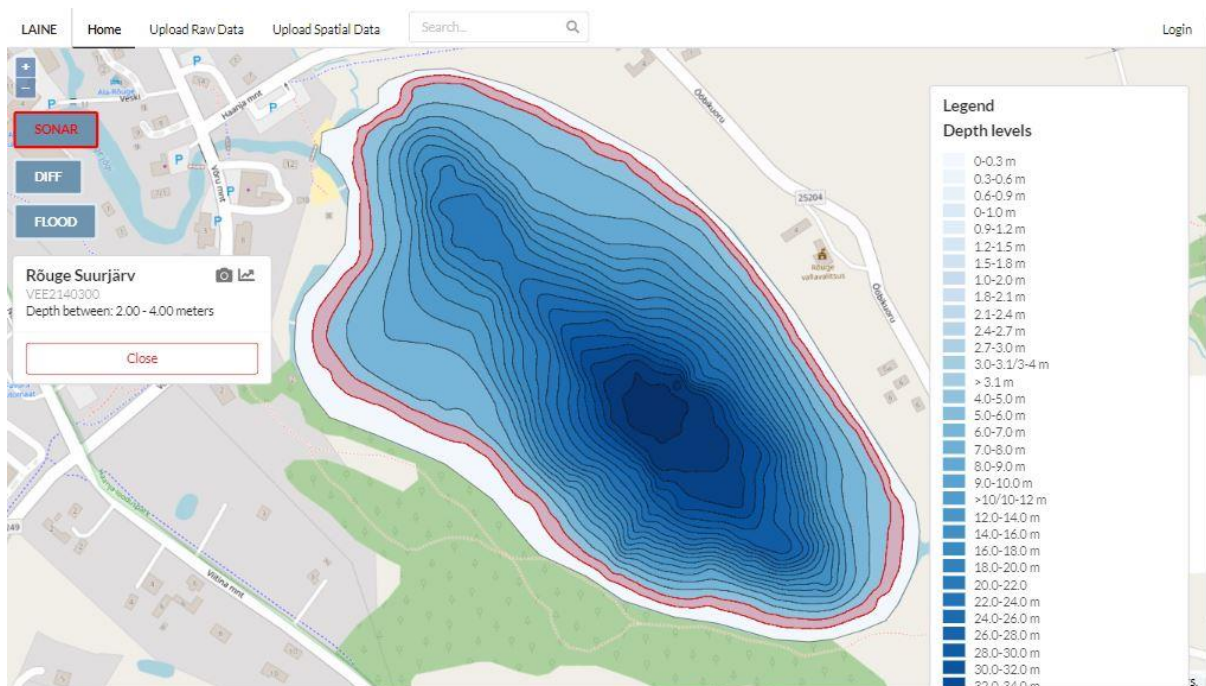
    HttpClientService.executeRequest(request, response);
}

```

Joonis 11. RESTful POST päring

5.5. Klientrakendus

Prototüüplahenduse klientrakendus pakub kasutajaliidest toorandmete laadimiseks andmetöötluse tarbeks ning toodetud kaardikihtide kuvamist ja infopäringuid. Samuti on võimalik visualiseerida veekogu põhjast tehtud 3D vaateid ning läbilõike profile.



Joonis 12. LAINE sügavusandmed

Klientrakendus programmeeriti kõige viimasena ning oli töö autorile omamoodi huvitav väljakutse, kuna kogu kaardikomponent tuli täiesti nullist üles seada.

Prototüübi klientrakenduse programmeerimise juures oli kõige keerulisem Openlayers-i seadistamine vastavaks L-EST projektsioonile, et loodud kaardikihid ja kaardivaated oleksid geograafiliselt õiges kohas. Samuti osutus ootamatuks väljakutseks kaardikihtide uuendamine, eelkõige selle seadistamine React-i töötamise loogika põhjal.

```
import proj4 from "proj4";
import { register } from "ol/proj/proj4";
import { get as getProjection } from "ol/proj";
...
proj4.defs(
  "EPSG:3301",
  "+proj=lcc +lat_1=59.33333333333334 +lat_2=58 +l
at_0=57.51755393055556 +lon_0=24 +x_0=500000 +y_0=6375000 +e
llps=GRS80 +towgs84=0,0,0,0,0,0 +units=m +no_defs"
);
register(proj4);
this.lEST = getProjection('EPSG:3301');
this.vectorLayer;
this.view;
...
this.view = new View({
  projection: this.lEST,
  center: [606426.34, 6512073.19],
  zoom: 8
});
...
```

Joonis 13. Aluskaardi seadistamine L-EST97 koordinaatsüsteemi

Joonisel 13 on väljavõte koodiridadest *Map.js* klassis, mis näitab aluskaardi seadistamist tasapinnalisse ristkoordinaatide süsteemi L-EST97. See on vajalik, et loodud kaardikihid asuksid „õige“ koha peal. Vaikimisi kasutab OpenLayers *Web Mercator* projektsiooni (kasutavad: *OpenStreetMap-project*, *Bing Maps*, *Google Maps*), kuid loodud LAINE kaardikihid on Eesti põhikaardi projektsioonis L-EST97.

6. Arutelu ja järeldused

Käesoleva töö tulemusena saadi klient-serverrakenduse prototüüp (tasemel TRL4/TRL 5), mis võimaldab luua kogutud sonariandmetest veekogu sügavuskaardi nii 2D kui 3D vaates, võrrelda sõltumatuid veekogu põhja kõrgusmudeleid ning simuleerida erinevaid veetasemeid.

Loodud lahendus ei vasta kõigile töö alguses püstitatud eesmärkidele, nagu näiteks veekogu põhja kõlvikukaardi või põhja profiilist koos tuvastatud objektidega 3D-pildi loomine. Veekogu kõlvikukaarti ei saanud valmistada paljuski seetõttu, et käesolevatest andmetest ei olnud võimalik ilma sonari tootjapoolse abita kätte saada vastavat infot. Põhja profiilist tuvastatud objektidega 3D-pildi loomiseks on aga tarvis külgvaatesonarit. Samuti ei ole GIS-töövood automatiseeritud, kuna töö autori käsutuses olevatele andmetele tuginedes ei leitud ühtegi igale veekogu punktipilvele sobivat universaalset töökäiku. Kui protsess sobiski ühtedele veekogu andmetele, siis teise veekogu andmetega tekkis kuskil mingi viga ja lõpptulemus oli vale.

6.1. Püstitatud eesmärkide täitmine

Töö alguses püstitatud eesmärk oli tootearenduse idee „veekogude põhjakaardi tootmise platvormi“ tehnilise teostatavuse uuring ja prototüüpimine kuni TRL 5 tasemel. Platvorm peab võimaldama automaatselt või vähese inimsekkumisega luua aluskaardi nii sügavusandmetega kui veekogu põhja füüsikaliste omadustega (kõlviku kaart), tuvastada objekte ning luua koos nendega põhja profiilist 3D-pilt. Samuti peaks rakendus suutma võrrelda sõltumatuid veekogu põhja kõrgusmudeleid ja simuleerida erinevaid veetasemeid. Lisaks loodava tarkvara arendamisele plaanis autor lasta ettevõttel AS Datel hinnata ka rakenduse ärilist tasuvust: mis väärtust see loob kliendile ja kas ettevõttel tasub antud toodet arendada.

Tabelis nr 3 on välja toodud funktsionaalsete ärinõuete rahuldamise võrdlus.

Tabel 3. Lähteülesandes esitatud funktsionaalsete ärinõuete rahuldamise võrdlus

Ärinõue	Sisaldab	Ei sisalda
Kliendil on võimalik määrata ulatust, milles ta veekogu kaardistust soovib	Jah*	Ei*
Aluskaardi loomine: sügavuskaart	Jah, lahendus kirjeldatud punktis 5.3.	
Aluskaardi loomine: põhja pinnakatte erisuse kaart		Ei, põhjendus punktis 5.3.
Aluskaardi loomine: objektide võimalik tuvastamine		Ei, põhjendus punktis 5.3.
Aluskaardi loomine: põhja 3D profiili genereerimine	Osaliselt (põhjast 3D pildi loomine), põhjendus punktis 5.3.	
Rakendus, mis võimaldab toodetud aluskaardi andmeid sirvida	Osaliselt (2D andmeid)	
Sirvimisrakendus võimaldab veetaseme muutuseid emuleerida	Jah, lahendus kirjeldatud punktis 5.3.	
Sirvimisrakenduses on võimalus sisse laadida ka sõltumatuid 3D mudeleid (kas varasemaid mõõtmisi või näiteks projekteeritud tammi põhja) ning teostada nende võrdlust (näiteks arvutada kubatuuri muutuseid)	Osaliselt (GIS tarkvarasse on võimalus sisse laadida sõltumatuid 3D mudeleid) ja teostada nende võrdlust.	

* Kliendil ei ole võimalik ise kaardile joonistada ulatust, kuid on võimalik anda punktandmed ainult soovitud piirkonna kohta või anda kaasa polügoon (*shape* fail või koordinaadid), mille ulatuses kaardistust soovib. Selle alusel on võimalik GIS rakenduses kaardistuse ulatus määrata.

Mittefunktsionaalsetest ärinõuetest täideti osaliselt:

1. Meie andmekogumise ja –tootmise platvorm on arendatud sõltumatult konkreetsest füüsilisest liikurplatvormist, kuid testitud vähemalt ühel liikuvplatvormil – prototüüp ei sõltu ühestki füüsilisest liikurplatvormist (sisendandmed on vajalikud teksti formaadis), kuid testida liikurplatvormidel ei jõutud.
2. On loodud demo kasutatavast platvormist (näiteks Rummu Karjääri põhjakaardi näitel osas, kus on veelune hoonestus) – prototüüplahendusest on loodud klient-serrakenduse näol demo, kuid andmed prototüübis pärinevad töö autori erakogust.

Mittefunktsionaalsetest ärinõuetest ei täidetud veel:

1. Me suudame orienteeruvalt hinnata tehnilise platvormi toorandmete kogumiseks ning lõppkasutajale soovitud andmete tootmiseks vajalikku aega.

Kokkuvõtvalt võib öelda, et loodud prototüüplahendusega saavutati lähteülesande nõuete punktide 3, 5, 6, 7, 8, 9, 10 täielik või osaline rahuldamine. See tagas ka ettevõtte AS Dateli huvi antud projektiga jätkamise osas.

6.2. Töö rakendamisest töökohal

Kohtumisel ettevõtte AS Datel esindajatega (müügiinimesed, intelligentsete transpordisüsteemide (ITS) valdkonna juht ja tarkvara osakonna juht) demonstreeriti prototüüplahendust ja arutati võimalusi, kuidas loodud lahendust kõige paremini ära kasutada. Leiti, et on mõttekas tootearendusega edasi minna, sest juba arendatud võimekuse juures on võimalik loodavate aluskaartide abil kliendile väärtust pakkuda. Samuti andis loodud demonstratsioon müügiinimestele julguse antud lahendust klientidele pakkuda ning ITS valdkonna juht nägi potentsiaali LAINE rakendust kasutada oma projektides.

Prototüüplahenduse järgmised planeeritud tööd:

- Luua veekogu põhjast 3D mudeli vaatamise võimekus veebirakendusse.
- Automatiseerida loodud GIS-töövood (võimekusega käivitada neid läbi rakenduse kasutajaliidese).
- Luua veekogu aluspõhja kõlviku kaart (eeldab andmekomplekti, mis sisaldab selleks vajalikke lähteandmeid).
- Luua veebirakenduse kasutajaliidesesse ruumiandmete töötlemise/muutmise võimekus.

Kokkuvõte

Käesolev töö on osa ettevõtte AS Datel tootearendusidee projektist LAINE ning kogu projektist kaetakse lõputöös projekti ettevalmistuse ja prototüübi arendamise osad. Tööst valmib eraldi ka magistritöö, mis hõlmab endas äriarhitektuuri ja -analüüsi (äri- ning süsteemianalüüs). Töö alguses püstitatud eesmärk oli tootearenduse idee „veekogude põhjakaardi tootmise platvormi“ tehnilise teostatavuse uuring ja prototüüpimine kuni TRL 5 tasemel.

Autor annab ülevaate tootearenduse ja prototüübi arendamise protsessi meetodikatest ja töövoogudest. Seejärel tutvustatakse põgusalt sarnaseid ja juba turul eksisteerivaid teenuseid/tooteid. Tehakse ülevaade milliseid arendusmetoodikaid, tehnoloogiaid ning tarkvara arenduse mustreid antud prototüübi loomisel kasutati ning põhjendatakse tehtud valikuid. Samuti kirjeldatakse kogu projekti loomist nõuete kogumisest ja arhitektuuri loomisest kuni prototüübi loomise protsessini välja.

Prototüübi madala TRL taseme (TRL 5) tõttu arvestab antud töö ISO/IEC 25010:2011 nõuete kategooriast ainult funktsionaalse sobivusega ja selle alamkategooriatega (funktsionaalne kohasus, funktsionaalne täielikkus ja funktsionaalne õigsus) ning käsitleb ITIL-i tavade ja protsesside juures teenuse kavandamise ja realiseerimise osa. Kavandamise protsessi osas täpsemalt disaini koordineerimist ning realiseerimise protsessis nõuete ja arhitektuuri haldamist.

Projekti alguses koostas töö autor koos meeskonnaliikme Siima Tiitusega esialgsed eepikud, *Lean Canvas*'e, *Requirements Discovery Canvas*'e, MVP ja liftikõne. Eelnimetatud materjale kasutas lähtematerjalina projekti meeskonnaliige Veikko Danilas projektile analüüsi koostamiseks ja oma magistritöö „Mehitamata seadmete vahendusel teostatava veekogude põhja seire ja kaardistuse võimekuse analüüs“ tarbeks. Saadud ärinõuete prioritseerimine toimus MoSCoW meetodil ja projekti tööde jaotamiseks kasutati WBS tehnikat.

LAINE platvormi realiseerimiseks kasutatavad tehnoloogiad on valitud lähtudes sellest, et ettevõttes AS Datel kasutab töö autor antud tehnoloogiaid oma igapäevatöös ning kuna loodava töö tulemus on lähtepunkt edasistele arendustele, siis rakenduse realiseerimisel on ettevõtte meeskonnal olemas antud tehnoloogiatel kompetents. Platvormi põhitehnoloogiad on Java, Javascript, PostgreSQL, GRASS GIS ja Geoserver.

Töö tulemusena saadi prototüüplahendus (tasemel TRL 4/TRL 5), mis koosneb klient-serverrakendusest, andmebaasist, geoserver ja GRASS GIS rakendusest. Andmekihtide ettevalmistamine toimub GRASS GIS nimelises rakenduses, kust valmiskihid laetakse otse Postgresql andmebaasi. Geoserver-rakendust kasutatakse ruumiandmete jagamiseks ja kuvamiseks WMS- ja WFS-teenustena. Serverrakendust kasutatakse andmebaasi ja geoserveriga suhtlemiseks ning klientrakendust andmete sisestamiseks, muutmiseks ja kuvamiseks.

Käesolev lahendus võimaldab luua kogutud sonariandmetest veekogu sügavuskaardi nii 2D kui 3D vaates, võrrelda sõltumatuid veekogu põhja kõrgusmudeleid ning simuleerida erinevaid veetasemeid. Kõiki töö alguses püstitatud eesmärke täita aga ei suudetud, nagu näiteks veekogu põhja kõlvikukaardi või põhja profiilist koos tuvastatud objektidega 3D-pildi loomine. Samuti ei ole GIS-töövood automatiseeritud, kuna töö autori käsutuses olevatele andmetele tuginedes ei leitud ühtegi igale veekogu punktipilvele sobivat universaalset töökäiku. Kui protsess sobiski ühtedele veekogu andmetele, siis teise veekogu andmetega tekkis kuskil mingi viga ja lõpptulemus oli vale.

Summary

Objective of this paper was to study the technical feasibility of the product development idea “ monitoring and mapping platform for water bodies” and to create a prototype of it up to TRL level 5. This work is part of the product development project LAINE for the company AS Datel and in this paper only project preparation and prototype development are covered. There is also a master's thesis that is completed separately, which includes business architecture and analysis (business and systems analysis).

Due to the low TRL level of the prototype (TRL 5), this work considers only functional suitability and its subcategories (functional suitability, functional completeness and functional accuracy) from the requirements category of ISO/IEC 25010:2011. It also addresses ITIL practices but only service design and implementation at processes. In terms of the design process, more specifically the coordination of design and the management of requirements and architecture in the implementation process.

At the beginning of the project, the author of the paper together with the team member Siima Tiitus compiled the initial epics, Lean Canvas, Requirements Discovery Canvas, MVP and elevator call. The aforementioned materials were used as a source material by Veikko Danilas, a member of the project team, to compile an analysis of the project and for his master's thesis “Analysis of the capacity of bottom monitoring and mapping of water bodies through unmanned equipment”. The obtained business requirements were prioritized using the MoSCoW method and the WBS technique was used to distribute the project work.

The technologies used for the implementation of the LAINE platform have been selected based on the fact that in AS Datel the author uses these technologies in his daily work and since the result of the work is a starting point for further developments, the company team has competence in these technologies. The main technologies of the platform are Java, Javascript, PostgreSQL, GRASS GIS and Geoserver.

The work resulted in a prototype solution (level TRL 4 / TRL 5), which consists of a client-server application, a database, a geoserver and a GRASS GIS application. The preparation of the data layers takes place in an application called GRASS GIS, from where the finished layers are loaded directly into the Postgresql database. The geoserver application is used as a WMS and WFS service to share and display spatial data. The server application is used to

communicate with the database and the geoserver, and the client application is used to enter, modify and display data.

This solution makes it possible to create a water body depth map from the collected sonar data in both 2D and 3D views, to compare independent water bottom elevation models and to simulate different water levels.

Kasutatud kirjandus

Applications architecture. (2021). *Wikipedia*. https://en.wikipedia.org/wiki/Applications_architecture

Ashwini, A. (2017). *What is the Difference Between React.js and React Native?*.
<https://www.cognitiveclouds.com/insights/what-is-the-difference-between-react-js-and-react-native/>

Banga, S. (2021). *What is Web Application Architecture? Components, Models, and Types*.
<https://hackr.io/blog/web-application-architecture-definition-models-types-and-more>

Bolat, S. (2014). *Technology Readiness Level (TRL) Math for Innovative SMES*.
<https://serkanbolat.com/2014/11/03/technology-readiness-level-trl-math-for-innovative-smes/>

Cohn, A. (2004). *User Stories Applied for Agile Software Development*. Pearson Education, Inc., Boston.

Concepts. *Webpack veebiportaal*. <https://webpack.js.org/concepts/>

Daityari, S. (2021). *Angular vs React vs Vue: Which Framework to Choose in 2021*.
<https://www.codeinwp.com/blog/angular-vs-vue-vs-react/>

Earth. (2021). *Wikipedia*. <https://en.wikipedia.org/wiki/Earth>

Fielding, R., T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. University of California, Irvine.

Foote, K. (2008). Underwater acoustic technology: review of some recent developments. *A Review in Oceans 08 MTS/IEEE Conference*.

Fowler, M. (2003). *Patterns of Enterprise Application Architecture*. Pearson Education, Inc., Boston.

Ghahrai, A. (2019). *Difference Between Iterative and Incremental Development in Agile*.
<https://devqa.io/iterative-incremental-development-agile/>

Gierke, O., Darimont, T., Strobl, C., Paluch, M. & Bryant, J. (2021). *Spring Data JPA – Reference Documentation*. <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/#reference>

GRASS. *OSGeo veebiportaal*. <https://grass.osgeo.org/learn/overview/>

Geoserver. *OSGeo veebiportaal*. <https://www.osgeo.org/projects/geoserver/>

Herkül, K. (2014). *Sonarisiüsteemi rakendamise meetodika loomine merepõhja elupaikade ja füüsiliste omaduste kaardistamiseks*. Tartu Ülikooli Eesti Mereinstituut, Tallinn.

Information architecture. (2021). *Wikipedia*. https://en.wikipedia.org/wiki/Information_architecture

Introduction to Java TM Technology. *Oracle*. <https://www.oracle.com/java/technologies/introduction-to-java.html>

ISO/EIC 25010:2011. *ISO veebiportaal*. <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>

Kaaser A. J., Litts T. L. (2013). *An illustrated Guide to Low-cost, Side Scan Sonar Habitat Mapping*.

Kaugseire. (2020). *Vikipeedia*. <https://et.wikipedia.org/wiki/Kaugseire>

MoSCoW Prioritisation. *Agilebusiness.org*.

https://www.agilebusiness.org/page/ProjectFramework_10_MoSCoWPrioritisation

MVC. *MDN Web Docs*. <https://developer.mozilla.org/en-US/docs/Glossary/MVC>

OpenLayers. *OSGeo veebiportaal*. <https://www.osgeo.org/projects/openlayers/>

PostGIS. *OSGeo veebiportaal*. <https://www.osgeo.org/projects/postgis/>

Riphyak, Y. (2017). *The Lean Canvas: how to fill it in and what should be taken into consideration*.

<https://wiki.rademade.com/lean-canvas>

Robinson, P. (2015). *The Requirements Discovery Canvas in a Nutshell*.

<https://lonsdalesystems.com/post/the-requirements-discovery-canvas-in-a-nutshell>

Roosaare, J., Mõisja, K. & Aunap, R., 2019. *Geoinformaatika*. Tartu Ülikool, Tartu.

Semantic UI React. *Semantic UI React veebiportaal*. <https://react.semantic-ui.com/>

Simmonds, J., MacLennan, D. (2005). *Fisheries Acoustics: Theory and Practice*. Blackwell Publishing, Oxford.

Sommerville, I. (2016). *Software Engineering, 10th edition*. Pearson Education, Harlow.

Spring Framework Documentation. *Spring veebiportaal*. <https://docs.spring.io/spring-framework/docs/current/reference/html/index.html>

spring initializer. *Spring veebiportaal*. <https://start.spring.io/>

Standardipõhine tarkvaratehnika sõnastik. <https://stats.cyber.ee/#showlist>

Tammemets, P. *Kasutajakeskne disain ja prototüüpimine. Peatükk 8: Prototüübi esitlemine*. Tallinna Ülikool. <https://web.htk.tlu.ee/digitalu/disain/chapter/prototuuubi-esitlemine/>

Tarkvaraprotsess_tegevused_2018. *Õppeaine IFI6231.DT Tarkvaraarenduse projekt materjal*.

http://www.cs.tlu.ee/~inga/TTP/Tarkvaraprotsess_tegevused_2018.pdf

Tepandi, J. (2020). *Tarkvara protsessid ja kvaliteet*. <https://tepani.ee/tns-loeng.pdf>

The Crossing The Chasm Elevator Pitch Template. *ElevatorPitchEssentials*.

<https://www.elevatorpitchessentials.com/essays/CrossingTheChasmElevatorPitchTemplate.html>

TIOBE indeks. *TIOBE*. <https://www.tiobe.com/tiobe-index/>

Tõnisson, H., Intervjuu e-posti vahendusel Tallinna Ülikooli Ökoloogia Instituudi Maastikuökoloogia osakonna teaduriga. Intervjueeris A.Porman (25.04.2010).

What is PostgreSQL?. *Postgresql veebiportaal*. <https://www.postgresql.org/docs/13/intro-what-is.html>

What is a Work Breakdown Structure?. *Workbreakdownstructure.com*.
<https://www.workbreakdownstructure.com/>

Lisa 1 - Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

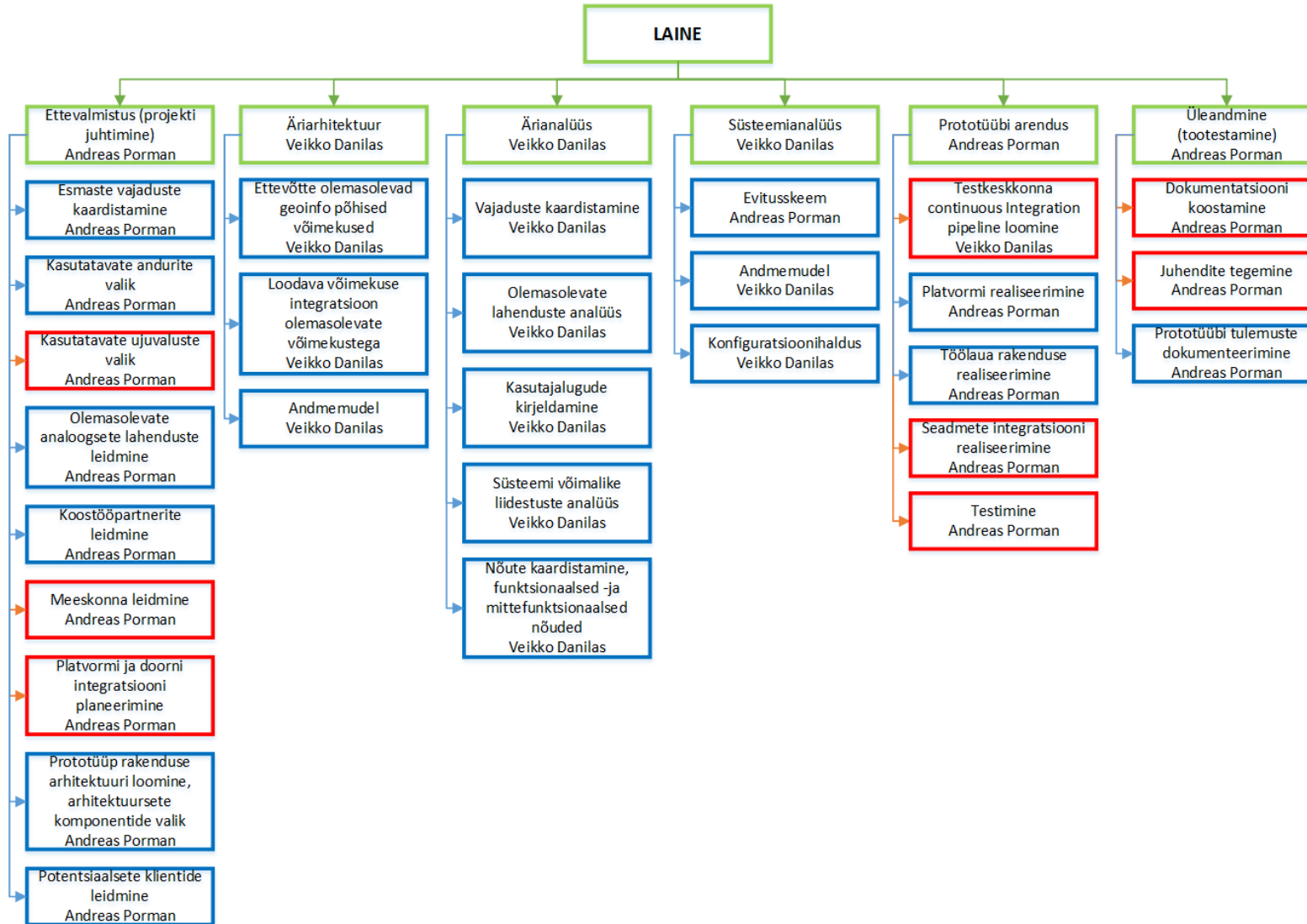
Mina, Andreas Porman

1. annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose " Veekogude põhja seire ja kaardistuse platvormi prototüüp“, mille juhendajad on Toomas Lepikult ja Agu Leinfeld
 - 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.

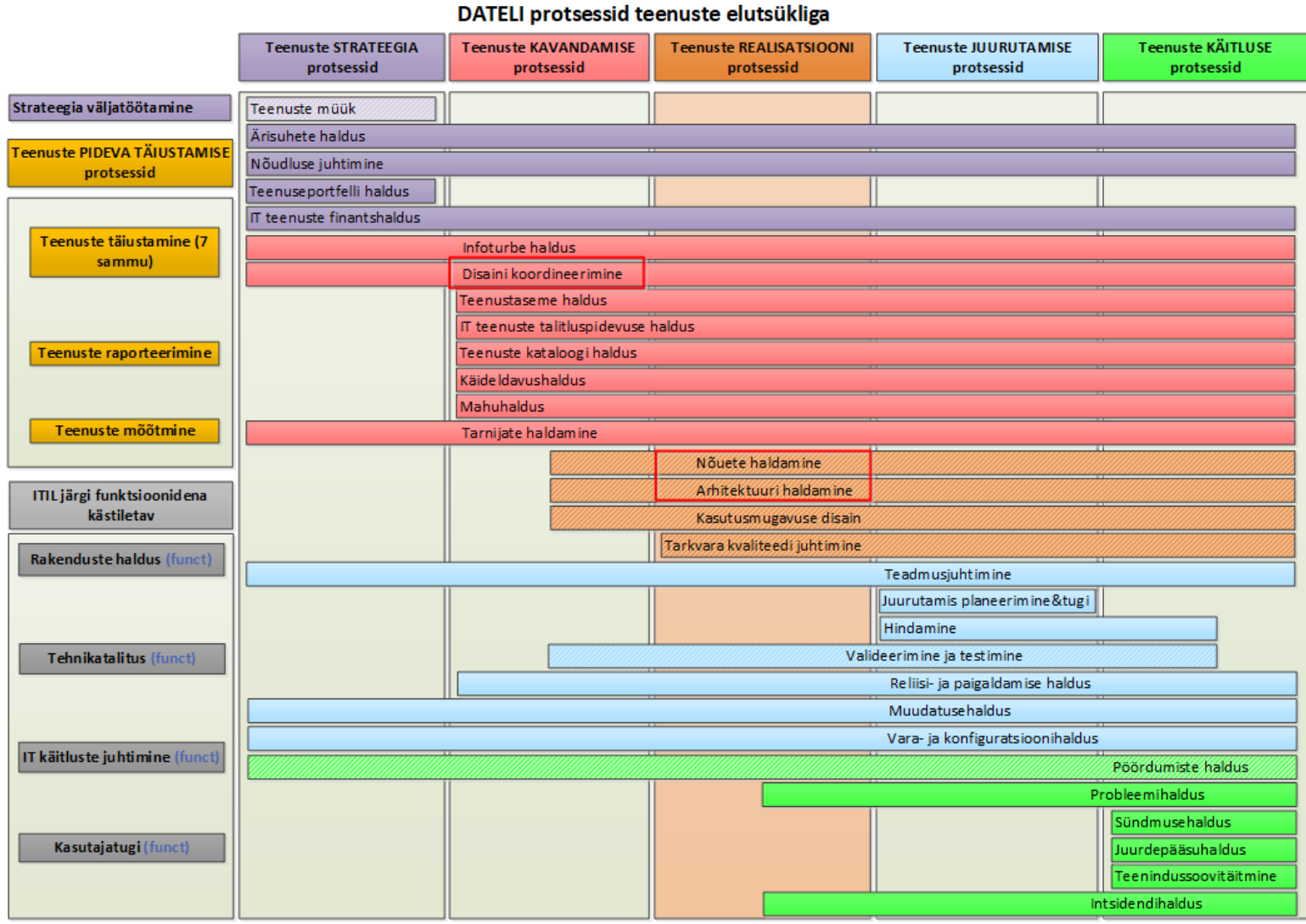
28.04.2021

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 loominguilise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2., siis lihtlitsents nimetatud tähtjaja jooksul ei kehti.

Lisa 2 – WBS



Lisa 3 – ITIL-i Dateli protsessid teenuste elutsükliga



Lisa 4 – TRL tasemed (Bolat, 2014)



Technology Readiness Levels

- TRL 0: Idea.** Unproven concept, no testing has been performed.
- TRL 1: Basic research.** Principles postulated and observed but no experimental proof available.
- TRL 2: Technology formulation.** Concept and application have been formulated.
- TRL 3: Applied research.** First laboratory tests completed; proof of concept.
- TRL 4: Small scale prototype** built in a laboratory environment ("ugly" prototype).
- TRL 5: Large scale prototype** tested in intended environment.
- TRL 6: Prototype system** tested in intended environment close to expected performance.
- TRL 7: Demonstration system** operating in operational environment at pre-commercial scale.
- TRL 8: First of a kind commercial system.** Manufacturing issues solved.
- TRL 9: Full commercial application,** technology available for consumers.

Lisa 5 – Lean ja Requirements Discovery Canvas näited

LEAN CANVAS Title: WATERBOT Created By: Andreas ja Siimo Date: 05.03.2020

PROBLEM List your top 3-5 problems. <ul style="list-style-type: none"> • AEGAMUDEV • OHTLIK • RESSURSI KULUDAS (INIM) 	SOLUTION Outline a possible solution for each problem. <ul style="list-style-type: none"> • KIIRENDAD OTSIMIST • MUUDAB TURVALISEMMA • VÄHENDAB INIMRESSURSI KULU 	UNIQUE VALUE PROPOSITION Single, clear, compelling message that states why you are different and worth paying attention. AUTONOOMNE OBJEKTIDE OTSIMINE ja TUUSTAMINE	UNFAIR ADVANTAGE Something that cannot easily be bought or copied. PLATUORM	CUSTOMER SEGMENTS List your target and users. • PAASTE AMET • POLITSEI
EXISTING ALTERNATIVES List how these problems are solved today. • INITOO TUNNID	KEY METRICS List the key numbers that tell you how your business is doing. • LIITSENTSIODE HOUK • KASUM	HIGH LEVEL CONCEPT List your X for Y analogy (e.g. YouTube - Flickr for videos) • PLATUORM OBJEKTIDE OTSIMISEKS • PLATUORM OBJEKTIDE TUUSTAMISEKS	CHANNELS List your path to customer (inbound or outbound) • OTSERAHNAMISEO • KOUURENTS • C2C (SÖBRALE SÖBRALE)	EARLY ADOPTERS List the characteristics of your ideal customers. • MURELIK TÖUKER • MURELIK REMISSERID
COST STRUCTURE List your fixed and variable costs. • TÖÖ TUNNID (MEESIOUNA KULUD) • HOOLDOOSTUNNID • DRÖÖN • ANDUUBID	REVENUE STREAMS List your sources of revenue. • TERVIKLAHEVUSE MÜÜK (LITSENTS) • HOOLDETEENUS • AREVOSTEENUS			

LEAN CANVAS is adapted from The Business Model Canvas (BusinessModelGeneration.com)
 LEANSTACK

Vajadus tuvastamiseks ja äärmise lahendamise jaoks

requirementsdiscoverycanvas-1.pdf • v.1 [Current] Edit with ↕

Requirements Discovery Canvas Prepared By: WATERBOT (Andreas & Siimo) Date: 19.05.2020

Business	Need	Solution
Product Owner Who is responsible for controlling the scope of the solution and ensuring that it provides value to the organization as a whole? Päärakend (Kadri Tomming)	Strategic Which strengths need to be built on or preserved, weaknesses remedied, opportunities exploited and/or threats avoided? Turvulik platvorm Mõju kasutajale Ajaühikute hõlmatus Muudatuste tuvastamine	Vision Who is the target customer for the solution? Kõik need, kes soovivad teha tööd ja on valmis maksama selle eest. Kõik need, kes soovivad teha tööd ja on valmis maksama selle eest. Kõik need, kes soovivad teha tööd ja on valmis maksama selle eest. Kõik need, kes soovivad teha tööd ja on valmis maksama selle eest.
Subject Matter Experts Who possesses expert knowledge of the problem domain or some aspect of the solution? Päärakend DATEL TÜ	Information What information needs to be stored, retrieved and manipulated? Pildid Videod Põhimi info (kõikvõrre info) Vahetugi info Kasutaja info	Interfaces What are the major user interfaces and APIs provided by the solution? Töölaadid Kõigilistele kasutajatele Põhimi kasutajale
Other Stakeholders Who else is involved (need to comply, need to consult, need to inform, need to restrict flow of information)? SMT	Rules What aspects of the problem domain need to be defined, constrained or guided in some way? Seadusandlus Sihtsüsteemi funktsioonid Andmevahetuse funktsioonid Andmetöötluse funktsioonid	Components What are the major components of the solution? Andmebaasi server Rakendusserver Veebiserver
Activities What activities do the stakeholders perform and what goals do they achieve by performing these activities? Töökoostöö Dröoni juhtimine Seadistamine Tööde haldamine Ohtuotsustamine Ohuotsustamine Andmete kogumine Andmete haldamine Andmete vahetamine	Features How can the solution satisfy a need by implementing a capability such as storing and retrieving data, interacting with users, interfacing with other systems and/or devices, or enforcing a rule? Kõigile info Edastada info Tööde info Visualiseerida info Automaatselt manuaalne info töötlus Tööplaandi haldamine Dröoni haldamine Andmete vahetamine Andmevahetamine	Product Owner Who is responsible for controlling the scope of the solution and ensuring that it provides value to the organization as a whole? Päärakend (Kadri Tomming)

Lisa 6 – Tootarendusfaasi esimesed eepikud

1. Töölaua kasutamine
 - 1.1. Autentimine ja autoriseerimine
 - 1.2. Välja logimine
2. Drooni haldamine - (Lähteülesanne punkt 8 ja 9)
 - 2.1. Drooni seadistamine
 - 2.2. Drooni logid
3. Drooni autonoomsus - (Lähteülesanne punkt 8 ja 9)
 - 3.1. Etteantud marsruudi järgimine
4. Seadmed - (Lähteülesanne punkt 8 ja 9)
 - 4.1. Anduri(te) seadistamine
5. Rakenduse veahaldus
 - 5.1. Vealogid
 - 5.2. Andmevahetuslogid - kui toimub andmete saatmine üle mobiilse andmeside
6. Tööde haldamine - (Lähteülesanne punkt 2)
 - 6.1. Uue töö loomine
 - 6.2. Töö muutmine
 - 6.3. Töö vaatamine
 - 6.4. Drooni ülesande loomine
 - 6.5. Tööde lisamine tööplaani
 - 6.6. Tööplaani muutmine
 - 6.7. Tööplaani vaatamine
7. Tööalade haldamine - (Lähteülesanne punkt 1)
 - 7.1. Uue tööala loomine
 - 7.2. Tööala muutmine
 - 7.3. Tööala vaatamine
8. Otsingud
 - 8.1. Tööde otsimine
 - 8.2. Tööplaanide otsimine
 - 8.3. Tööalade otsimine
 - 8.4. Seadmete otsimine
 - 8.5. Objekti otsimine
9. Andmete kogumine - (Lähteülesanne punkt 3a ja 3b)

- 9.1. Salvestamine
 - 9.1.1. Sügavusandmed
 - 9.1.2. Tagasihajuvuse intensiivsus
 - 9.1.3. Nõlvakalle
 - 9.1.4. Konarlikkus
 - 9.1.5. Terasuurus
- 10. Andmete töötlemine - (Lähteülesanne punkt 4)
 - 10.1. Sonari logi töötlus
- 11. Andmete visualiseerimine - (Lähteülesanne punkt 5)
 - 11.1. Kaart - (Lähteülesanne punkt 4)
 - 11.2. Tabel
 - 11.3. Graafik
 - 11.4. Andmete eksport
 - 11.5. Aruanne/raport
- 12. Andmevahetus väliste süsteemidega - (Lähteülesanne punkt 5)
 - 12.1. WMS ja WFS teenused

Lisa 7 – MoSCoW meetodil prioritseeritud lähteülesanne

Must have:

1. Kliendil on võimalik määrata ulatust, milles ta veekogu kaardistust soovib.
3. Me oleme võimelised looma aluskaardi vähemalt järgmiste andmetega:
 - 3.1. Sügavusandmed;
 - 3.2. Põhja pinnakatete eristus (n.ö veealune kõlvikute kaart).
4. Meil on rakendus, mis võimaldab kogutud toorandmetest automaatselt või vähese inimsekkumisega toota punktis 3 loetletud veepõhja kaardi.
5. Meil on rakendus, mis võimaldab toodetud aluskaardi andmeid sirvida.

Should have:

2. Me suudame orienteeruvalt hinnata tehnilise platvormi poolt toorandmete kogumiseks ning lõppkasutajale soovitud andmete tootmiseks vajalikku aega.
8. Meie andmekogumise ja –tootmise platvorm on arendatud sõltumatult konkreetsest füüsilisest liikurplatvormist, kuid testitud vähemalt ühel liikuvplatvormil.
9. On loodud demo kasutatavast platvormist (näiteks Rummu Karjääri põhjakaardi näitel osas, kus on veealune hoonestus).
10. Meil on ülevaade, millised on konkureerivate platvormide teenuse osutamise aeg, hind ning nende pakutava andmete detailsus (millised kategooriad) ja kvaliteet (lahutus/ täpsus) (sh kasutame ära suvel kogutud infot platvormide ja teenuste kohta).

Could have:

3. Me oleme võimelised looma aluskaardi vähemalt järgmiste andmetega:
 - 3.3. Erihuviobjektide võimalik tuvastamine (näiteks suuremad kivid, vrakid, metall objektid, veealused tehisrajatised);
 - 3.4. Võimalusel põhja profiili 3D-pildi genereerimine.
6. Sirvimisrakendus võimaldab veetaseme muutuseid emuleerida.
7. Sirvimisrakenduses on võimalus sisse laadida ka sõltumatuid 3D-mudeleid (kas varasemaid mõõtmisi või näiteks projekteeritud tammi põhja) ning teostada nende võrdlust (näiteks arvutada kubatuuri muutuseid).

Won't have:

-

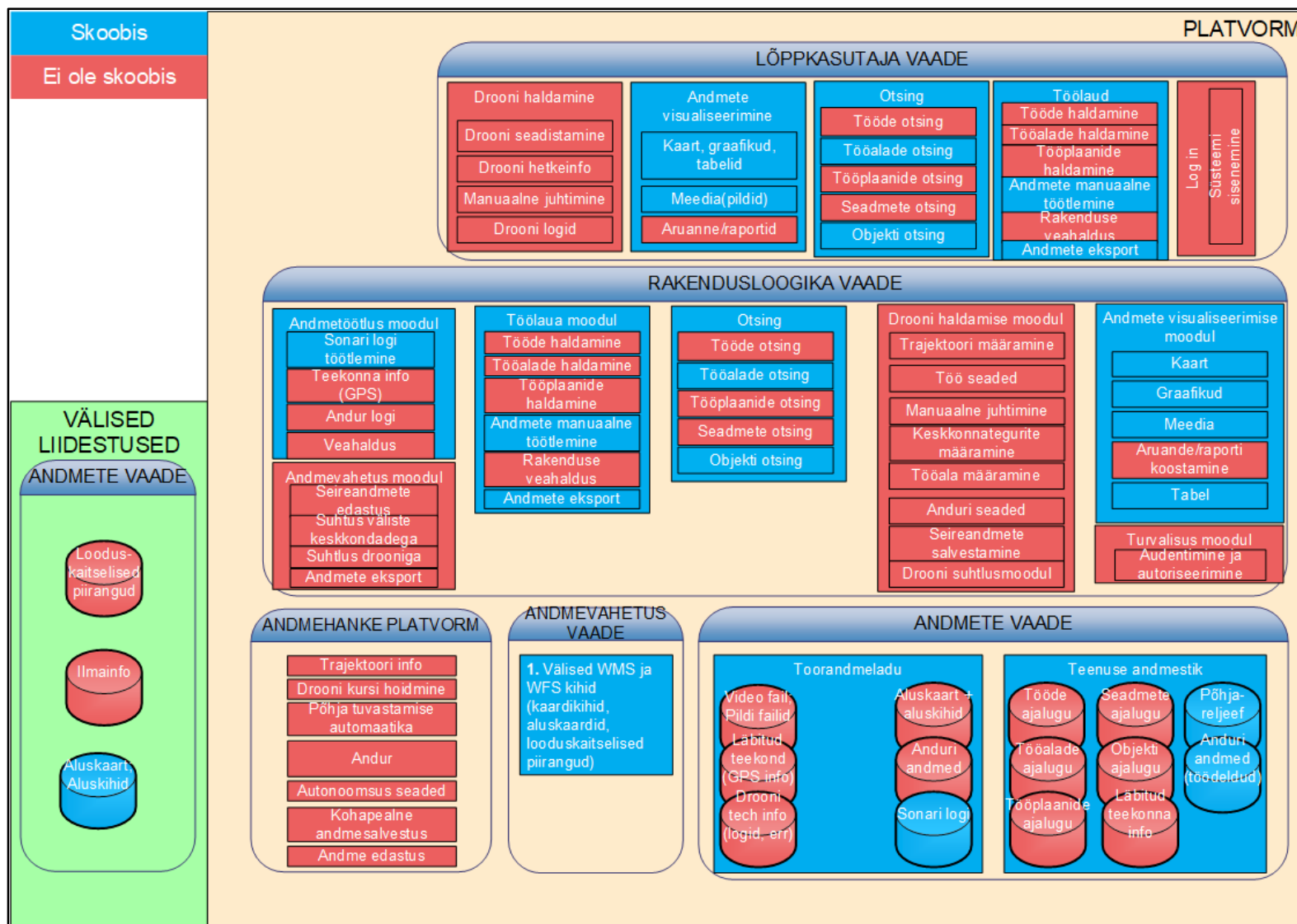
Lisa 8 – Võrdlus turul olevate konkureerivate toodete/teenustega

Ärinõuded / tooted	Bio Base Maps	Garmin ActiveCaptain	R-Süsteemid
Aeg	6h*	-	-
Hind	330 – 2321 EUR	Tasuta**	-
Kliendil on võimalik määrata ulatust, milles ta veekogu kaardistust soovib.	+	-	+
Aluskaardi loomine: sügavusandmed	+	+	+
Aluskaardi loomine: põhja pinnakatte eristus	+	-	-
Aluskaardi loomine: objektide võimalik tuvastamine (näiteks suuremad kivid, vrakid, metall objektid, veealused tehisrajatised).	-	-	+
Põhja profiili 3D pilti genereerimine.	-	-	+
Rakendus, mis võimaldab toodetud aluskaardi andmeid sirvida.	+	+	+
Sirvimisrakendus võimaldab veetaseme muutuseid emuleerida.	-	-	-
Sirvimisrakenduses on võimalus sisse laadida ka sõltumatuid 3D mudeleid (kas varasemaid mõõtmisi või näiteks projekteeritud tammi põhja) ning teostada nende võrdlust (näiteks arvutada kubatuuri muutuseid).	-	-	-
Andmekogumise ja –tootmise platvorm on sõltumatu konkreetsest füüsilisest liikurplatvormist.	-	-	+

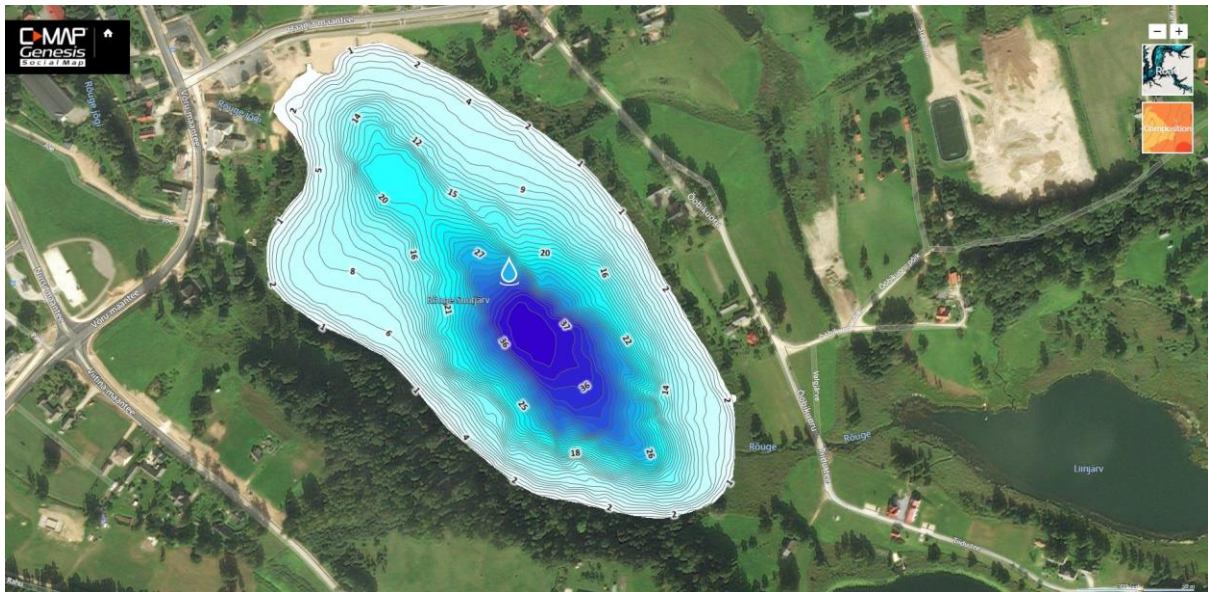
* ligikaudne aeg, mis kulus veebirakendusse sonari logi faili (Lauga järve näitel) laadimisest kuni teateni, et andmetest on aluskaardid moodustatud.

** veebikaardi rakenduses andmete sirvimine

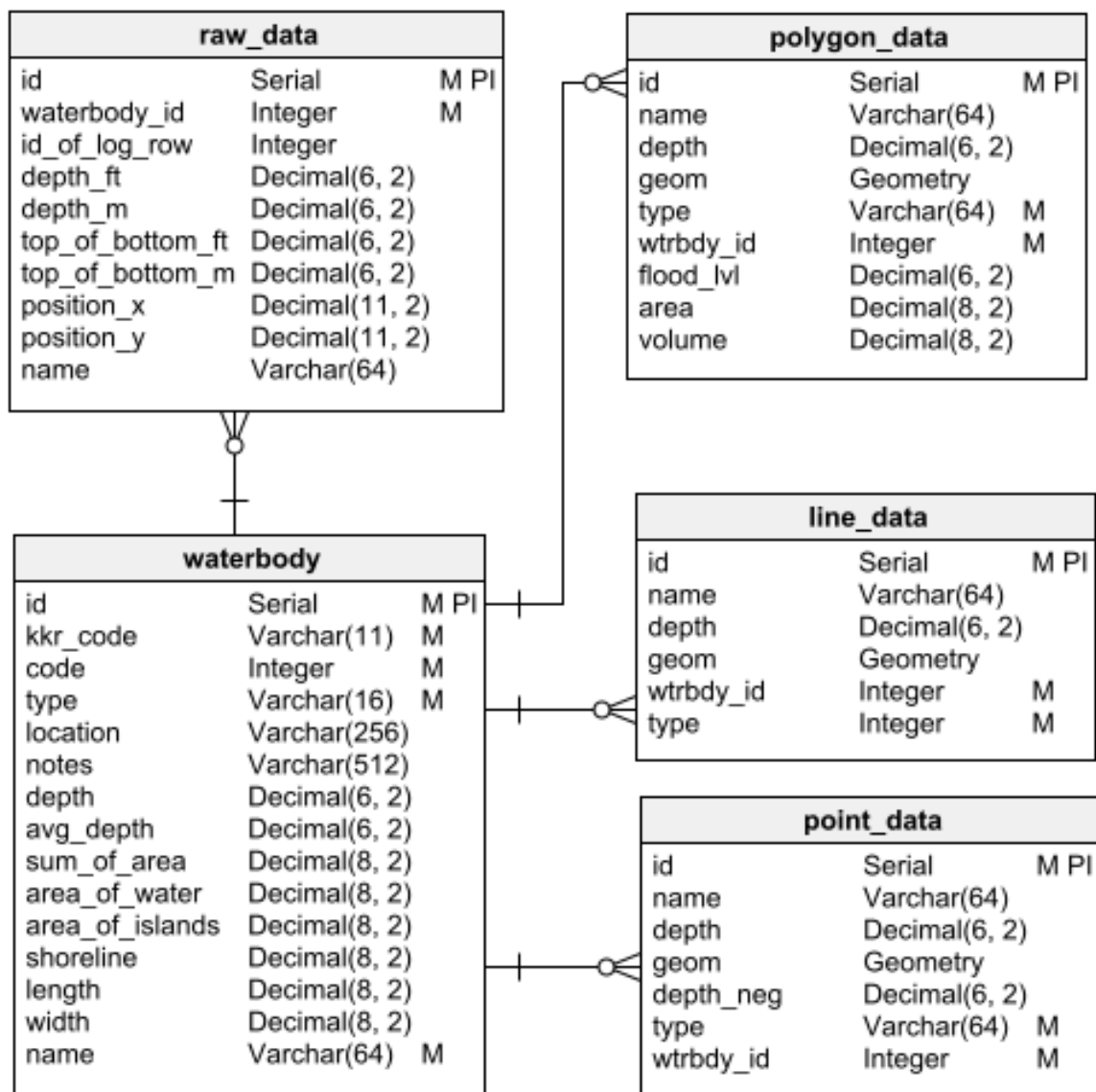
Lisa 9 – LAINE kommunikatsiooni arhitektuur



Lisa 10 – C-Map Genesis sügavuskaart (kuvatõmmis C-Map Genesis)



Lisa 11 – LAINE ERD mudel



Lisa 12 – Sonarist saadud toorandmed

	UpperLimit	LowerLimit	DepthValid	Depth	WaterTempValid	WaterTemp	Temp2Valid	Temp2	Temp3Valid	Temp3	WaterSpeedValid	WaterSpeed	PositionValid	PositionX	Positi												
1	0.000000	32.799999	T	9.880000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884228	T	1.291927	T	9.876302	F	T	80	T	0.738661	6.248279	F	0.000000
2	0.000000	32.799999	T	9.840000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884228	T	1.291927	T	9.844271	F	T	196	T	0.738661	6.248279	F	0.000000
3	0.000000	32.799999	T	9.790000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884228	T	1.291927	T	9.790885	F	T	314	T	0.738661	6.248279	F	0.000000
4	0.000000	32.799999	T	9.770000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884228	T	1.291927	T	9.769531	F	T	435	T	0.738661	6.248279	F	0.000000
5	0.000000	32.799999	T	9.760000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.758854	F	T	561	T	0.738661	6.230825	F	0.000000
6	0.000000	32.799999	T	9.750000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.281250	T	9.748177	F	T	697	T	0.738661	6.230825	F	0.000000
7	0.000000	32.799999	T	9.800000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.281250	T	9.801562	F	T	912	T	0.738661	6.230825	F	0.000000
8	0.000000	32.799999	T	9.790000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.281250	T	9.790885	F	T	1021	T	0.583153	6.230825	F	0.000000
9	0.000000	32.799999	T	9.750000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.748177	F	T	1156	T	0.583153	6.230825	F	0.000000
10	0.000000	32.799999	T	9.700000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.281250	T	9.694792	F	T	1271	T	0.583153	6.230825	F	0.000000
11	0.000000	32.799999	T	9.670000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.673437	F	T	1397	T	0.583153	6.230825	F	0.000000
12	0.000000	32.799999	T	9.620000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.281250	T	9.620051	F	T	1506	T	0.583153	6.230825	F	0.000000
13	0.000000	32.799999	T	9.590000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.588020	F	T	1641	T	0.583153	6.230825	F	0.000000
14	0.000000	32.799999	T	9.560000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.555989	F	T	1767	T	0.583153	6.213372	F	0.000000
15	0.000000	32.799999	T	9.560000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.555989	F	T	1793	T	0.583153	6.213372	F	0.000000
16	0.000000	32.799999	T	9.490000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.281250	T	9.491927	F	T	1925	T	0.583153	6.213372	F	0.000000
17	0.000000	32.799999	T	9.450000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.281250	T	9.449219	F	T	2041	T	0.252700	6.213372	F	0.000000
18	0.000000	32.799999	T	9.430000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.427864	F	T	2165	T	0.252700	6.213372	F	0.000000
19	0.000000	32.799999	T	9.380000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.385156	F	T	2292	T	0.252700	6.213372	F	0.000000
20	0.000000	32.799999	T	9.350000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.353125	F	T	2419	T	0.252700	6.213372	F	0.000000
21	0.000000	32.799999	T	9.320000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.321094	F	T	2548	T	0.252700	6.213372	F	0.000000
22	0.000000	32.799999	T	9.270000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.267708	F	T	2682	T	0.252700	6.213372	F	0.000000
23	0.000000	32.799999	T	9.230000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.224999	F	T	2799	T	0.252700	6.230825	F	0.000000
24	0.000000	32.799999	T	9.200000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.203646	F	T	2929	T	0.252700	6.230825	F	0.000000
25	0.000000	32.799999	T	9.190000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.192968	F	T	3063	T	0.252700	6.230825	F	0.000000
26	0.000000	32.799999	T	9.180000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.182291	F	T	3176	T	0.252700	6.230825	F	0.000000
27	0.000000	32.799999	T	9.180000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.182291	F	T	3298	T	0.252700	6.230825	F	0.000000
28	0.000000	32.799999	T	9.170000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.171615	F	T	3407	T	0.252700	6.230825	F	0.000000
29	0.000000	32.799999	T	9.140000	T	6.850006	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.139584	F	T	3533	T	0.116631	6.230825	F	0.000000
30	0.000000	32.799999	T	9.080000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.075521	F	T	3651	T	0.116631	6.230825	F	0.000000
31	0.000000	32.799999	T	9.010000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.291927	T	9.011458	F	T	3858	T	0.116631	6.230825	F	0.000000
32	0.000000	32.799999	T	8.950000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.217188	T	8.947395	F	T	3981	T	0.116631	6.230825	F	0.000000
33	0.000000	32.799999	T	8.930000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.217188	T	8.926042	F	T	4100	T	0.116631	6.230825	F	0.000000
34	0.000000	32.799999	T	8.880000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.206510	T	8.883333	F	T	4220	T	0.602592	6.195919	F	0.000000
35	0.000000	32.799999	T	8.830000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.206510	T	8.829947	F	T	4336	T	0.602592	6.195919	F	0.000000
36	0.000000	32.799999	T	8.850000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884229	T	1.206510	T	8.851302	F	T	4461	T	0.602592	6.195919	F	0.000000
37	0.000000	32.799999	T	8.900000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884230	T	1.206510	T	8.904687	F	T	4586	T	0.602592	6.195919	F	0.000000
38	0.000000	32.799999	T	8.910000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884230	T	1.206510	T	8.904687	F	T	4711	T	0.602592	6.195919	F	0.000000
39	0.000000	32.799999	T	8.910000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884230	T	1.206510	T	8.904687	F	T	4731	T	0.602592	6.195919	F	0.000000
40	0.000000	32.799999	T	8.910000	T	6.869995	F	0.000000	F	0.000000	F	0.000000	T	2986577	7884230	T	1.206510	T	8.904687	F	T	4731	T	0.602592	6.195919	F	0.000000