

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Anna Linskaja 175183IDDR

# **Analüüsiseadmete integreerimine ainete tuvastamise rakendusega**

Diplomitöö

Juhendaja: Evelin Halling  
PhD

Kaasjuhendaja: Jelena Gorbatšova  
PhD

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: Anna Linskaja

17.05.2021

## Annotatsioon

Käesoleva bakalaureusetöö eesmärgiks on täiendada Tallinna Tehnikaülikooli Keemia ja biotehnoloogia instituudis projekti, mis on seotud kapillaarelektroforeesil põhinevate analüütiliste seadmetega. Täpsemalt oli töö ülesandeks analüüsida ja võrrelda erinevate laboriseadmete väljundfaile ja luua nende jaoks rakendus, mis oskab töödelda iga seadme katsetulemuste väljundfaile ja loob töötlemisel leitud andmete põhjal uue JSON formaadis faili, mis postitatakse REST API kaudu ainete tuvastamise veebirakendusse luues sinna uue eksperimendi sissekande.

Töö tulemusena valmis Python programmeerimiskeeles kirjutatud rakendus teadlaste laborikatsete tulemuste töötlemiseks ja veebirakendusse edastamiseks, kus on võimalik valida: mis tüüpi seadme failidega tegu on, valida eksperimendi tüüpi ja sisestada korrutajat, mis on vajalik signaali väärtuste korrektseks edastamiseks. Iga käivitamisel rakendus küsib vajaliku sisendi protsessimiseks lisainformatsiooni; töötleb läbi kõik etteantud kaustas olevad failid ja kataloogid; kogutud informatsiooni pealt loob kasutaja katseandmete kaustas eraldi „\_Out“ kausta, kuhu salvestab JSON formaadis välja saatmiseks valmis failid; edastab failid veebirakendusse. Loodud rakendus võimaldab kasutada ka ainult veebirakendusse postitamise funktsionaalsust.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 28 leheküljel, 6 peatükki, 18 joonist, 1 tabelit.

## **Abstract**

### **Integration of Analyzers into Substance Detection Application**

The aim of this bachelor's thesis is to supplement a project of the Institute of Chemistry and Biotechnology of Tallinn University of Technology, what is related to analytical equipment based on capillary electrophoresis. More specifically, the main task of the work was to analyze and compare experiment result's output files for different laboratory devices and create an application for them, what can parse each device experiment result file and create a JSON that is posted via the REST API to the substance detection web application, creating a new experiment entry there.

As a result of the thesis, an application written in the Python programming language is ready for use in production, what can parse the results of the researchers' laboratory experiments and can post them to a web application. Each time the application is launched it: asks for additional information needed to process the input: what is the device type, experiment type and multiplier, what is needed for correct signal data posting; parses through all files and folders, what are in the same same folder with the application; on top of the collected information, it creates and saves separate „\_Out“ folder in the user's folder with experiments, where are ready to be sent files in JSON format; transfers files to web application. The created application also allows to use only the posting functionality to the web application.

The thesis is in Estonian and contains 28 pages of text, 6 chapters, 18 figures, 1 table.

## Lühendite ja mõistete sõnastik

analüüt	testitavas proovis olev aine (komponent)
API	<i>Application Programming Interface</i> , rakendusliides
BGE	<i>Background Electrolyte</i> ehk taustelektrolüüt lahusesse lisatavad analüüdid, näiteks äädikhape, mille tagajärjel muutuvad elektroferogrammil tekkinud <i>peak</i> 'id üksteisest eristatavaks
CE	Capillary Electrophoresis ehk kapillaarelektroforees
<i>Debugging</i>	Silumine on tarkvaraarenduse protsess, mille käigus leitakse ja parandatakse vigu koodis, mis takistavad programmi või süsteemi korrektset toimimist. [1]
file-based	süsteemid, mida kasutatakse andmefailide korrastamiseks ja hooldamiseks
JSON	<i>JavaScript Object Notation</i> , andmevahetusvorming
JVM	<i>Java Virtual Machine</i> , virtuaalmasin (abstraktne masin) mis interpreteerib kompileeritud Java baitkoodi arvuti protsessorile arusaadavateks instruktsioonideks
JWT	<i>JSON Web Token</i> , JSON veebitunnus
kapillaar	Kvarts-, klaas- või polümeer toru mille sisemine läbimõõt varieerub 25-100 $\mu\text{m}$ ja välimine läbimõõt varieerub 150-350 $\mu\text{m}$
kasutajakogemus	<i>User experience</i> , kasutajakogemus on subjektiivne kvaliteedimõõdik, kus kasutaja alateadlikult hindab süsteeme, mida ta kasutab [2]
metafile	fail meetodi algandmetega
meetod	Katsemeetod koos kogu seadistuse infoga
parsimine	andmete lugemine ning seejärel loetud andmetega mingite tegevuste sooritamine
<i>Peak</i>	piik ehk tipp, tekib elektroferogrammile, kui mõõdetava analüüdi kontsentratsioon on üle eeltingimused on korrektsed ning elektrivälja mõjul moodustub elektroferogrammile tipp
REST	<i>Representational State Transfer</i> , tarkvara arhitektuuri stiil

## Sisukord

1 Sissejuhatus .....	10
1.1 Kapillaarelektroforeesi mõiste.....	11
1.2 Kapillaarelektroforeesi seade .....	12
2 Probleemi püstitus .....	14
2.1 Probleemi täpsustus .....	14
2.2 Töö eesmärk .....	15
3 Analüüs.....	16
3.1 Analüüsi metoodika.....	16
3.2 Olemasolevate rakenduste ülevaade .....	17
3.2.1 Töölauarakendus.....	17
3.2.2 Veebirakendus .....	18
3.3 Kapillaarelektroforeesi seadmete elektroferrogrammide analüüs .....	19
3.3.1 Drug Hunter Analyzer .....	20
3.3.2 Agilent .....	22
3.3.3 CE-FLU2 .....	24
3.3.4 CE-C4D .....	26
3.4 Lahendusviiside võrdlus .....	28
3.5 Nõuded loodavale süsteemile .....	29
3.5.1 Nõuded loodavale rakendusele.....	29
4 Arenduse tulemus .....	31
4.1 Kasutatud metoodika .....	31
4.1.1 Microsoft Visual Studio .....	31
4.1.2 Versioonihaldus GitLab.....	31
4.1.3 Python.....	32
4.1.4 File-based andmesüsteem .....	32
4.1.5 JSON.....	32
4.2 Loodud rakendus .....	32
4.2.1 Koostöö veebirakendusega .....	33
4.2.2 Sisendandmete töötlus .....	33

4.3 Tegevusvoog.....	34
5 Testimine .....	37
5.1 Rakenduse manuaalne testimine.....	37
5.2 Kasutajakogemuse testimine .....	37
5.3 Automaattestimine.....	37
6 Kokkuvõte .....	38
Kasutatud kirjandus .....	39
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	40
Lisa 2 – Eksperimendi loomise mudel, mis on kasutatud /api/internal/experiments API sees .....	41
Lisa 3 – Drug Hunter Analyzer seadmest saadud XML failist katse sagedus ja pinged..	47
Lisa 4 – Rakenduse sisendfailide võrdlus.....	49

## Jooniste loetelu

Joonis 1. Kapillaarelektroforeesi seade (seade on välja töötatud TalTech-s, KBI-s). Joonis publitseeritakse TalTech, KBI, Mikrofluidika keskuse labori teaduri Jelena Gorbatsova loaga. ....	10
Joonis 2. Elektroosmootne voog kapillaaris. [3] .....	11
Joonis 3. Skemaatiline kapillaarelektroforeesi seadme ehitus.....	13
Joonis 4. Eesmärgi visualiseerimine.....	15
Joonis 5. Veebirakenduses katsete listi vaade. ....	18
Joonis 6. Drug Hunter Analyzer seade. [13] .....	20
Joonis 7. Drug Hunter Analyzer väljundfailide list.....	21
Joonis 8. Drug Hunter Analyzer elektroferogrammi näite. ....	22
Joonis 9. Agilent 7100 CE System seade. [8].....	23
Joonis 10. Agilent seadme katsete väljundfailide list.....	23
Joonis 11. Agilent seadme elektroferogrammi näite. ....	24
Joonis 12. CE-FLU2 kapillaarelektroforeesi seade. Model - Kati Muldma, Fotograaf - Lauri Mölder. Publitseeritakse TalTech, KBI, teadurite Jelena Gorbatsova ja Jekaterina Mazina-Šinkar loaga.....	24
Joonis 13. CE-FLU2 seadme katsete väljundfailide list.....	25
Joonis 14. CE-FLU2 seadme elektroferogrammi näite. ....	26
Joonis 15. CE-C4D kapillaarelektroforeesi seade. Model - Kati Muldma, Fotograaf - Lauri Mölder. Publitseeritakse TalTech, KBI, teaduri Jelena Gorbatsova loaga. ....	27
Joonis 16. CE-C4D seadme katsete väljundfailide list.....	27
Joonis 17. Loodud konsoolirakendus. ....	33
Joonis 18. Rakenduse töövoog. ....	36



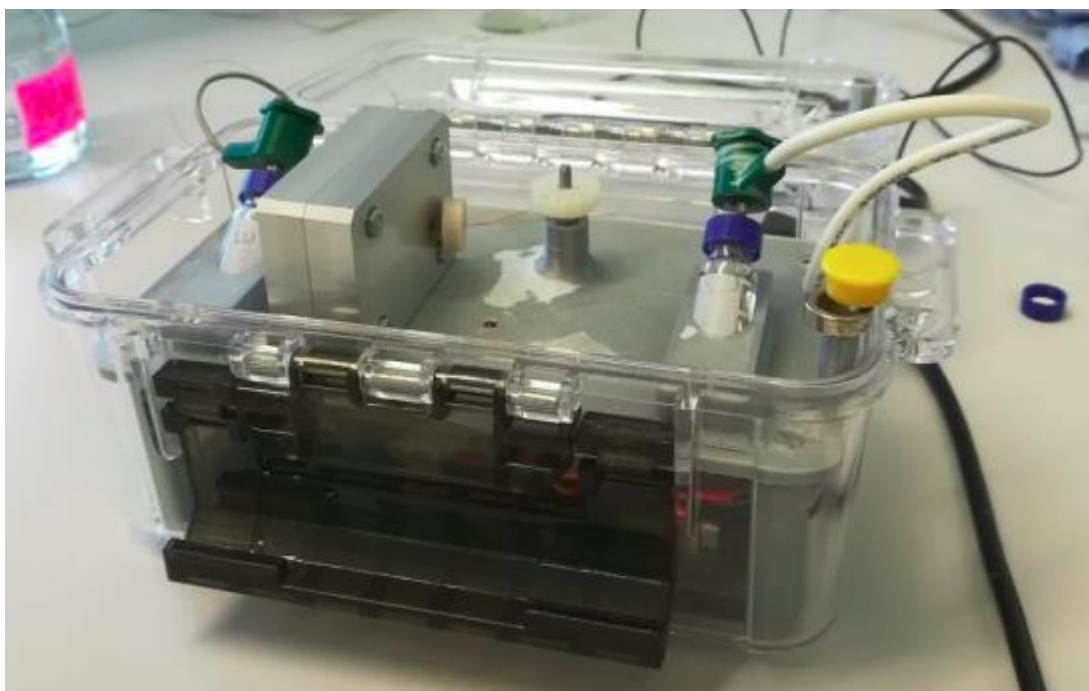
## **Tabelite loetelu**

Tabel 1. Programmeerimiskeelte võrdlus.....	28
---------------------------------------------	----

## 1 Sissejuhatus

Elektroforees on analüütide lahutamise meetod, kus laetud osakesed liiguvad elektrijuhtivust omavas vedelas keskkonnas elektrivälja mõjul ja mida kasutatakse erinevate keemiliste elementide ning ühendite tuvastamiseks. Elektroforeesi füüsilist protsessi kasutatakse täna erinevates tööstusharudes ja laialdaselt analüütilises keemias: DNA, RNA ja valgu analüüsimisel. [3]

Käesolevas lõputöös keskendutakse elektroforeesi alammeetodile: kapillaarsele elektroforeesile (Joonis 1) ja selle meetodi seadmetest tulnud katsete resultaatide edastamisele veebirakendusse, kus neid edasi analüüsitakse.

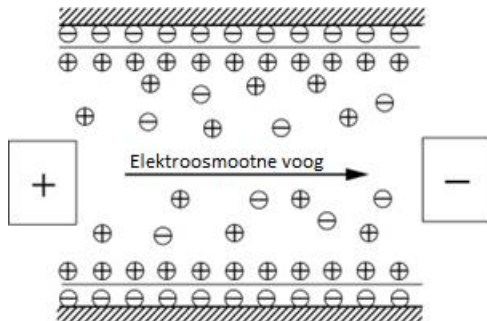


Joonis 1. Kapillaarelektroforeesi seade (seade on välja töötatud TalTech-s, KBI-s). Joonis publitseeritakse TalTech, KBI, Mikrofluidika keskuse labori teaduri Jelena Gorbatsova loaga.

## 1.1 Kapillaarelektroforeesi mõiste

Kapillaarelektroforeesis ehk KE (inglise keeles *Capillary Electrophoresis* ehk *CE*) on lahutusmeetod, mis põhineb laetud osakeste liikumisel erineva kiirusega elektrivälja mõjul kapillaarkolonnis. Antud meetodi arendati välja 1980. aastatel James Jorgenson ja Krynn Lukas. [4]

Kapillaarelektroforeesi korral viiakse elektroforeesi protsess läbi 25–100-mikromeetrise sisediameetriga kvarts-, klaas- või polümeerses kapillaaris, mis on täidetud taustelektrolüüdiga (*Background Electrolyte* ehk *BGE*) ja millele rakendatakse kõrgepinge 10–30 kV. Pärast seda, kui proov on sisestatud hakkavad selle komponendid elektriväljas elektroosmootse voo (Joonis 2) mõjul liikuma erineva kiirusega vastavalt osakeste laengule ja mõõtmetele, mistõttu toimub nende lahutamine tsoonideks. Detailselt: mida suurem on iooni laeng ja elektrivälja tugevus, seda kiiremini liigub ioon ja mida suuremad on iooni raadius ja puhvri viskoossus, seda suurem on takistusjõud. Kapillaari lõpuosas olevas detektoris registreeritakse komponentide kontsentratsioonile vastavad signaalid (piigid) ajalises järjestuses ning saadakse elektroferogramm. [3] [4]

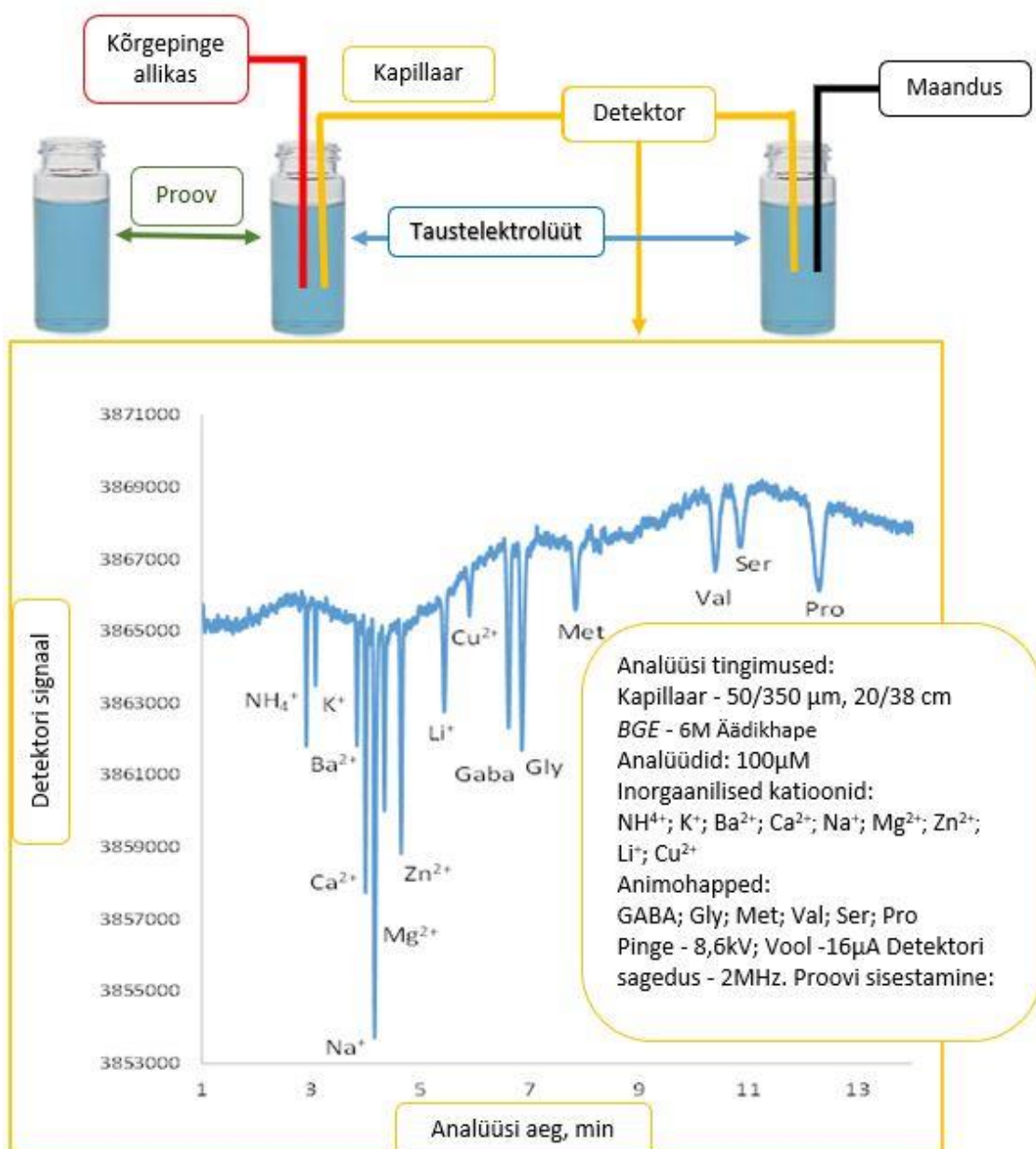


Joonis 2. Elektroosmootne voog kapillaaris. [3]

## 1.2 Kapillaarelektroforeesi seade

Kapillaarelektroforeesi seade (Joonis 3) on lihtsa ehitusega ja koosneb järgmistest osadest:

- reguleeritav kõrgepinge allikas;
- kaks anumad elektrolüüdi lahusega (puhverlahusega), mis asuvad ühel ja samal tasemel ja sisaldavad anoodseid ja katoodseid lahuseid;
- kaks elektroodi (katood ja anood), mis on paigaldatud igaüks eraldi puhverlahusega anuma sisse ja ühendatud toiteallikaga;
- kapillaar, milles toimub osakeste eraldamine;
- detektor, mis võimaldab kontrollida tuvastatud ainete arvu;
- termostaat, mis hoiab konstantset temperatuuri kapillaari sees;
- arvuti, mis registreerib eksperimendi tulemust. [5]



Joonis 3. Skemaatiline kapillaarelektroforeesi seadme ehitus.

## 2 Probleemi püstitus

Käesolevas bakalaureusetöös jätkatakse ja täiustatakse TalTech-i Keemia ja biotehnoloogia instituudis kapillaarelektroforeesil põhinevate analüütiliste seadmete projekti – luuakse programmeerimiskeeles Python kirjutatud rakendus, mille eesmärk on erinevatest analüüsiseadmetest väljastatud elektroferogrammide alusel moodustada uus JSON formaadis fail, mis postitatakse REST API kaudu ainete tuvastamise veebirakendusse luues sinna uue eksperimendi sissekande. JSON koostatakse sisendfailide põhjal, mis on iga seadmel erinevas formaadis, erineva struktuuri ja andmestikuga.

### 2.1 Probleemi täpsustus

Probleemiks, mis oli välja toodud Keemia ja biotehnoloogia instituudi teadlaste poolt ja mis leidis kinnitust töö analüüsi käigus on olukord, kus:

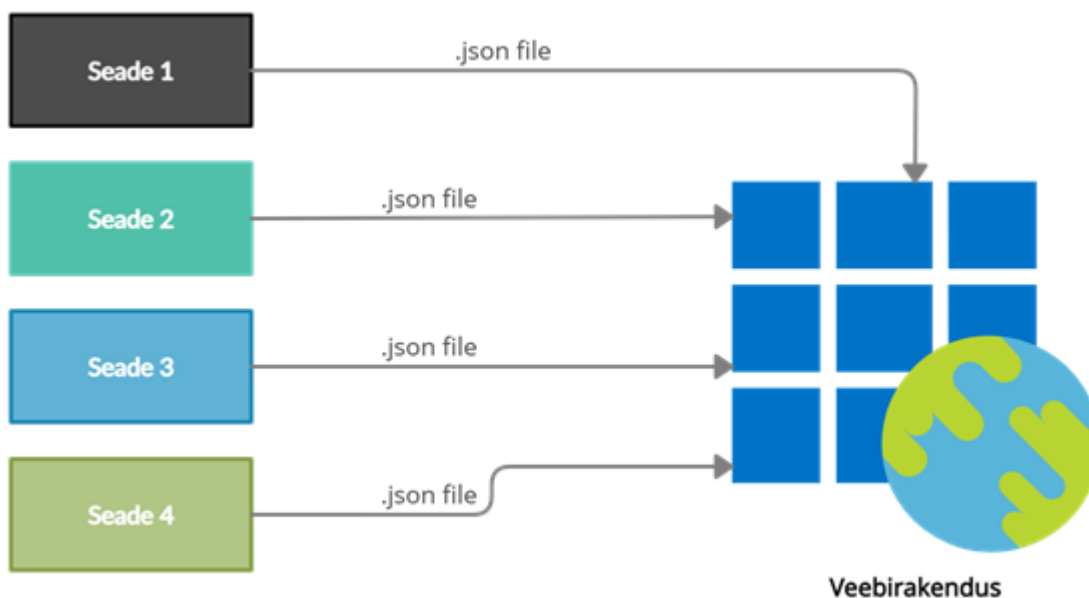
- on olemas veebirakendus, mis omab REST API otspunkti (i.k. endpoint) */api/internal/experiments* ja katse sisendandmeid võetakse vastu JSON vormingus
- kolm kapillaarelektroforeesi seadet salvestavad oma katsetulemusi erinevates formaatides ja erineval kujul ning nende tulemuste sisestamine veebirakendusse on manuaalne ja väga ajamahukas
- ainult ühel kapillaarelektroforeesi seadmel on olemas tööluarakendus, mis edastab katsetulemuste failid veebirakendusse automaatselt internetiühenduse olemasolul

Kliendi probleemi analüüsi käigus selgus, et kunagi oli loodud skript Python programmeerimiskeeles, kus oli lahendatud ühele kindlale kapillaarelektroforeesi seadmele järgmised tegevused: autentimine veebirakendusse, sisendfailidest andmete lugemine, JSON vormingus väljundfaili loomine ja andmete postitamine veebirakendusse. Antud skript polnud teadlaste poolt kasutusele võetud, vajab

lisaanalüüsi ja täiendamist, aga selle autentimise ja postitamise funktsionaalsust oleks võimalik taaskasutada rakenduse edasiarendamisel.

## 2.2 Töö eesmärk

Antud töö eesmärgiks on luua töötav rakendus, mis on võimeline looma neljast erinevast analüüsiseadmest väljastatud elektroferrogrammist JSON vormingus eksperimendi faili ja postitama selle ainete tuvastamise veebirakendusse (Joonis 4) - luues sinna uue eksperimendi sissekande. Rakendus peab sobima erineva tasemega kasutajatele ja olema lihtsasti hallatav. Rakenduse loomine põhieesmärk on kasutaja aja kokkuhoidmine ja andmete unifitseerimine vastavalt veebirakenduse sisendite parameetritele.



Joonis 4. Eesmärgi visualiseerimine.

## 3 Analüüs

Kolmas lõputöö peatükk keskendub analüüsile. Siin on kirjeldatud valitud analüüsi meetodikat ja selle valiku põhjendus. Tehakse ülevaade laborikatsete teostamiseks kasutatavatest rakendustest. Põhjalikult analüüsitakse erinevate kapillaarelektroforeesi seadmete elektroferogramme. Võrreldatakse alternatiivseid lahendusviise – programmeerimiskeele. Kirjeldatud on ka nõuded loodavale süsteemile, mida on kasutatud sisendiks rakenduse realiseerimisel.

### 3.1 Analüüsi meetodika

Antud bakalaureusetöös kasutatakse kvalitatiivse analüüsi meetodikat. Kvalitatiivne analüüs on selline analüüs, kus lähteandmed, nende töötlemine ja tehtud järeldused ei ole seotud arvuliste näitajatega. Tavaliselt kvalitatiivse uurimuse korral: püütakse vastata küsimustele: „Mis?“, „Miks?“ ja „Kuidas?“; tegeletakse sõnaliste karakteristikutega ja objektide kirjeldusega; uuritakse süsteeme neid jälgides. Kvalitatiivse analüüsi käigus keskendutakse: objekti süvaanalüüsile – lõputöö raames antud objekt on kapillaarelektroforeesi seadmete elektroferogrammid; uuritakse toimuva sisu – antud töö kontekstis veebirakendusse eksperimendi postitamise võimalust. [2]

Üks kvalitatiivse uurimuse tüüpidest on tegevusuuring, mille etapid on väga sobilikud just IT süsteemide arendamiseks ja missuguseid on enamasti kasutatud töö käigus:

- olukorra kaardistamine ja lähtekohtade selgitamine
- tegevuse või mõjutusprogrammi idee leidmine – ehk rakenduse läbimõtlemine
- tegevuse käivitamine ja selle levimisele kaasaaitamine – rakenduse arendamine ja käivitamine
- mõjutuse jälgimine ja tähelepanekute tegemine – testimine



- hindamine – testimise resultaate analüüs ja kasutaja tagasiside analüüs
- hilisem tegevus (tegevuse võimalik korrigeerimine ja juurutamine) – rakenduse korrigeerimine vastavalt testimise resultaate analüüsile

Kuna diplomitöö põhieesmärgiks oli olemasolevate seadmete väljundfailide analüüs, nende unifiitseerimine ja katsetulemuste edastamiseks rakenduse loomine, siis valikule jäi tegevusuuring.

## 3.2 Olemasolevate rakenduste ülevaade

Antud diplomitöö osas vaadeldakse olemasolevaid rakendusi ja nende töötamise põhimõtted. Veebirakenduse analüüs oli vajalik selleks, et aru saada, mis andmeid on vaja selleks, et luua eksperiment ja kuidas neid vastu võetakse.

### 3.2.1 Töölauarakendus

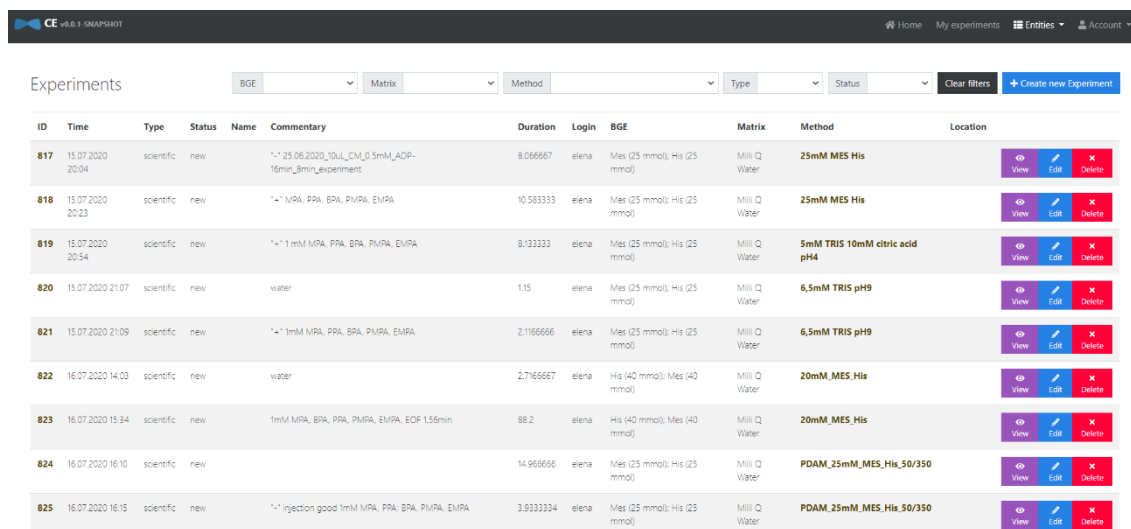
Töölauarakendus on programm, kus on võimalik alustada uut laborikatset ja seadistada selle parameetrid: katse tegija nimi, kapillaari mõõtmed, testitavad analüüdid, testitava analüüdi meetod ja selle sageduse muutmise plaan, testimiskeskond, elektrivoolu sagedus, kõrgepingeline ning selle protsentuaalne kasutus, katse kestus ja ka graafikul nähtav registreeritud signaal. Töölauarakendus on kasutusel ainult ühe kapillaarelektroforeesi seadmel – CE-C4D. Iga tehtava katse jooksul tekib algandmete, elektrivoolu tugevuse ja sageduse graafik, mis muutub reaalsajas.

Katseandmed salvestatakse vahemällu ja kui kasutaja otsustab, et välja tulnud katse rahuldab teda, siis ta saab salvestada andmed edasiseks kasutamiseks. Andmed salvestatakse kasutaja kettale ja veebirakendusse. Rakenduse seadetes on määratud kaust, mis luuakse kettale salvestamise hetkel. Iga katse kohta lisatakse kausta 4 faili: experiment.json, experiment\_image.png, human\_readable.txt ja raw.txt.

Kui on olemas internetiühendus, siis saadetakse experiment.json fail, katseandmete fail JSON kujul, läbi veebirakenduse API otspunkti süsteemi edasiseks analüüsiks. [6] [7]

### 3.2.2 Veebirakendus

Olemasolev veebirakendus on loodud JHipster platvormil selleks, et kasutajal oleks võimalik kapillaarelektroforeesi katsetes tekkinud andmeid töödelda (Joonis 5). Andmete tekitamine on võimalik läbi API ja veebirakenduse kasutajaliidese. Läbi kasutajaliidese on mugav luua katset ja katsekomponente ning süsteem on ka vähese kogemusega kasutajale arusaadav, aga samas see on väga ajamahukas, kuna informatsiooni maht, mida tuleb täita, on väga suur ja seoses sellega eksisteerib vea tekkimise oht. Eeltoodud põhjuse tõttu kasutajaliides sobib rohkem andmete parandamiseks ja muutmiseks. [7]



ID	Time	Type	Status	Name	Commentary	Duration	Login	BGE	Matrix	Method	Location
817	15.07.2020 20:04	scientific	new		*- 25.06.2020_10uL_CM_0.5mM_ADP-16min_8min_experiment	8.066667	elena	Mes (25 mmol); His (25 mmol)	Milli Q Water	25mM MES His	
818	15.07.2020 20:23	scientific	new		*- 1uPA, PPA, BPA, PMPA, EMPA	10.583333	elena	Mes (25 mmol); His (25 mmol)	Milli Q Water	25mM MES His	
819	15.07.2020 20:54	scientific	new		*- 1 mM MPA, PPA, BPA, PMPA, EMPA	8.133333	elena	Mes (25 mmol); His (25 mmol)	Milli Q Water	5mM TRIS 10mM citric acid pH4	
820	15.07.2020 21:07	scientific	new		water	1.15	elena	Mes (25 mmol); His (25 mmol)	Milli Q Water	6,5mM TRIS pH9	
821	15.07.2020 21:09	scientific	new		*- 1 mM MPA, PPA, BPA, PMPA, EMPA	2.116666	elena	Mes (25 mmol); His (25 mmol)	Milli Q Water	6,5mM TRIS pH9	
822	16.07.2020 14:03	scientific	new		water	2.716667	elena	His (40 mmol); Mes (40 mmol)	Milli Q Water	20mM MES His	
823	16.07.2020 15:34	scientific	new		1mM MPA, BPA, PPA, PMPA, EMPA, EOF 1.56min	88.2	elena	His (40 mmol); Mes (40 mmol)	Milli Q Water	20mM MES His	
824	16.07.2020 16:10	scientific	new			14.966666	elena	Mes (25 mmol); His (25 mmol)	Milli Q Water	PDAM_25mM MES_His_50/350	
825	16.07.2020 16:15	scientific	new		*- injection good 1mM MPA, PPA, BPA, PMPA, EMPA	3.933334	elena	Mes (25 mmol); His (25 mmol)	Milli Q Water	PDAM_25mM MES_His_50/350	

Joonis 5. Veebirakenduses katsete listi vaade.

Kõigi andmete salvestamiseks veebirakenduses on olemas API otspunkt `/api/internal/experiments`. Andmeid saadetakse Experiment objekti sees ning seejärel salvestatakse katsekomponente eri loogika järgi. Experiment klassi objekt – veebirakenduse põhiobjekt – katse. Iga katse tegemisel määratakse talle tüüp, staatus, aeg, kestus, kommentaar, uuritavad analüüdid (*analytesOfInterests*), mõõdetavad väärtused (*experimentResults*), kus kõige tähtsam on katses saadud signaali väärtus (*measuringPoint*). Lisaks on katsel olemas meetod (*method*), millega antud katse oli teostatud. Meetodil on omakorda enda parameetrid mida kasutatakse katse tegemisel. Need on: süstimisviisi tüüp, kapillaari tüüp, sagedus, süstimise kestus ja väärtus, kõrgepinge väärtus, kirjeldus, *BGE* ja analüüdid. API sisendina on katse oma andmetega JSON kujul. (Lisa 2)

Iga eelnimetatud parameeter on vajalik veebirakenduses analüüsi tegemisel, aga mõnedel parameetritel on suurem tähtsus kui teistel – nende sees on tüüp, staatus ja katses saadud

signaali väärtus, mille põhjal edaspidise analüüsi käigus tuvastatakse, mis ainega tegu oli. Signaali väärtuseid võetakse vastu veebirakenduse poolt täisarvuna ja see on punkt, millele on vaja tähelepanu pöörata. Praeguses süsteemis on olemas kaks eksperimendi tüüpi:

- *regulaar* (tavaline) - näitab, et tegu on tavalise laborikatsega ning selle katse tulemused ei täienda positiivsete katsete andmebaasi;
- *scientific* (teaduslik) - näitab, et katse tegija oli teadlane ning tehtud katse võib täiendada positiivsete katsete andmebaasi ja vajab edaspidist analüüsimist. [5]

Süsteemis eksisteerivate staatuse väärtuseid on kolm ja nad on:

- *new* (uus) - näitab, et tegemist on uue katsega, millele peab tähelepanu pöörama/analüüsima;
- *positive* (positiivne) - näitab, et katsetulemused täiendavad positiivsete katsete andmebaasi ja tehtud eksperimendi andmeid ja analüüsi hakatakse kasutama tulevastest arvutustes. Antud staatust määrab käsitsi teadlane;
- *negative* (negatiivne) - näitab, et tegemist oli vigase katsega, katse lisatakse negatiivsete katsete andmebaasi. Antud staatust määrab käsitsi teadlane.

### **3.3 Kapillaarelektroforeesi seadmete elektroferrogrammide analüüs**

Antud lõputöös olid põhjalikult analüüsitud nelja erineva kapillaarelektroforeesi seadme väljundfailid ja ühe seadme puhul oli analüüsitud ka väljundfaili taas-saatmise võimalust. Vaadeldud seadmed on järgmised:

- Drug Hunter Analyzer (TalTech, KBI-s välja töötatud)
- Agilent (kommerts toode)
- CE-FLU2 (TalTech, KBI-s välja töötatud)
- CE-C4D (TalTech, KBI-s välja töötatud)

Iga seade omab oma unikaalset väljundfailide komplekti. Väljundfailide formaadid on erinevad ja andmestik nende sees varieerub ka sõltuvalt seadme tüübist. Edaspidi on kirjeldatud põhjalikumalt igat seadet ja tema elektroferrogrammi tüüp.

### 3.3.1 Drug Hunter Analyzer















Drug Hunter Analyzer on kapillaarelektroforeesi seade fluorestsentsse detektoriga, mis oli loodud TalTech-i Keemia ja biotehnoloogia instituudi ja Siseministeeriumi koostöös (Joonis 6). Drug Hunter Analyzer on kaasaskantav analüsaator ebaseaduslike uimastite kuritarvitamise tuvastamiseks suukaudses vedelikus. Selle seadme abiga suukaudne vedelik näitab hiljutist narkootikumide tarbimist ja selle analüüsi tagajärjel on teada, mis ainet inimene tarbis.



Joonis 6. Drug Hunter Analyzer seade. [13]

Drug Hunter väljastab mitut erinevat tüüpi faile: csv ja txt (Joonis 7), mis salvestatakse ühe ja sama kausta sisse seadmega seotud arvuti kettale. Ühe kausta sees võib olla mitu erinevat katset, mis on tehtud sama kuupäeva, analüütide lahuse, meetodi, sageduse ja

maksimaalse pinge väärtusega. Seadmest on võimalik ka saada XML fail, mille sees on sagedus ja maksimaalse pinge väärtus. (Lisa 3)

	1_320091525_20x.csv		30.04.2021 11:58	Microsoft Excel C...	109 KB
	1_320091525_20x.txt		30.04.2021 11:58	Text Document	15 KB
	2_320101612_20x.csv		30.04.2021 11:58	Microsoft Excel C...	109 KB
	2_320101612_20x.txt		30.04.2021 11:58	Text Document	15 KB
	3_320101916_20x.csv		30.04.2021 11:58	Microsoft Excel C...	109 KB
	3_320101916_20x.txt		30.04.2021 11:58	Text Document	15 KB
	3_320101916_20x.xlsx		30.04.2021 11:58	Microsoft Excel W...	55 KB

Joonis 7. Drug Hunter Analyzer väljundfailide list.

CSV on kõige olulisem kapilaarelektroforeesi eksperimendi andmetega fail - elektroferogramm, selle sees on olemas sellised väärtused nagu: eksperimendi algus (*Start*) ja lõpp (*End*), möödunud aeg katse algusest (*timeFromStart*), laborikatse signaali väärtus (*PMTValue*), pinge (*voltage*), elektrivoolu andmed (*current*), surve (*pressure*) ja muud. Veebirakenduses katse edaspidisel analüüsil, teadlased vajavad antud failist ainult töödeldud eksperimendi algust (*Start*) ja lõppu (*End*), laborikatse signaali väärtust (*PMTValue*) ja elektrivoolu andmed (*current*). Signaali väärtuste täpsus võib katsetel erineda, aga sagedamini see on kaks kohta pärast koma (Joonis 8). Eksperimendi lõpust eksperimendi algust lahutades saab kätte eksperimendi kestuse, mis on vajalik edaspidise analüüsi jaoks. TXT failis on tehtud katse signaali andmed. Katse meetodi kirjelduse jaoks oli kunagi loodud JSON formaadis fail algandmetega – *metafile*, kuhu teadlased sisestavad ise: süstimisviisi tüüp, kapillaari tüüp, süstimise kestus ja väärtus, kõrgepinge väärtus, kirjeldus, *BGE*, analüüdid ja kommentaar. Analüüsi käigus selgus, et *metafile* oli loodud JSON formaadis selle tõttu, et veebirakendus võtab vastu andmeid ka antud formaadis. Kõik meetodi algandmeid on võimatu automaatselt kätte saada Drug Hunter

Analyzer seadmest, aga neid on vaja edaspidise eksperimendi analüüsi jaoks veebirakenduses.

	A	B	C	D	E	F	G	H	I	J	K
1	320092035										
2	Start	30.09.2020	0,86								
3	End	30.09.2020	0,87								
4	Counts	1521									
5											
6	Data	begin									
7	index	timeFromStart	PMTValue	refValue	PMTOk	refOk	current	voltage	pressure	temperature	
8	0	0,2357143	18319,43	288308,6	TRUE	TRUE	0,698394	38,35454	-0,05305756	0	
9	1	0,7119048	24030,87	266611,1	TRUE	TRUE	0,665423	35,40685	-0,02333709	0	
10	2	1,188095	24465,46	287364	TRUE	TRUE	3,230265	360,7118	0,06027013	0	
11	3	1,664285	24337,29	279816,5	TRUE	TRUE	11,35824	3952,835	0,06611904	0	
12	4	2,140476	24200,22	272317	TRUE	TRUE	15,99557	7126,852	0,02687397	0	
13	5	2,616667	23873,32	290672,9	TRUE	TRUE	21,09057	10248,24	0,05228693	0	
14	6	3,092857	23491,02	291525,1	TRUE	TRUE	26,29471	13348,43	0,07724853	0	
15	7	3,569048	23205,72	275218,3	TRUE	TRUE	31,69126	16435,88	0,08015475	0	
16	8	4,045238	22738,51	288579,7	TRUE	TRUE	37,33348	19505,06	0,05319498	0	
17	9	4,52143	22209,29	292513	TRUE	TRUE	43,34901	22563,72	0,03802311	0	
18	10	4,997619	21792,16	282843,2	FALSE	TRUE	46,85336	24956,66	0,03402605	0	

Joonis 8. Drug Hunter Analyzer elektroferogrammi näite.

### 3.3.2 Agilent

Kapillaarelektroforeesi seade vedelik kromatograafia fluorestsentsse detektoriga. Antud seade on laboratooriumis kasutamiseks, kuna ta on väga kompleksne (kaalub 35 kg) aparaat, mis on valmistatud Agilent Technologies poolt. See võimaldab automatiseerida katset, integreerida, kalibreerida, identifitseerida analüüte (Joonis 9). [8]



Joonis 9. Agilent 7100 CE System seade. [8]

Agilent seadmeist tulevad välja ainult csv formaadis failid, mis salvestatakse seadmega seotud arvuti kettale määratud kausta sisse. Ühe kausta sees võib olla mitu erinevat katse tulemust (Joonis 10), mis on teostatud samal kuupäeval, sama lahuse ja katse meetodiga. Meetodi fail antud seadmel puudub.

	3UG.CSV		13.04.2021 18:30	Microsoft Excel C...	82 KB
	10UG.CSV		13.04.2021 18:30	Microsoft Excel C...	60 KB
	15UG.CSV		13.04.2021 18:30	Microsoft Excel C...	57 KB

Joonis 10. Agilent seadme katsete väljundfailide list

CSV failis on antud seadme elektroferrogramm, mille sees on olemas laborikatse signaali väärtused (2. veerg failis) ja möödunud aja väärtused katse algusest (1. veerg failis). Signaali väärtuste täpsus võib erineda tehtud katsetel, aga sagedamini see on neliteist numbrit pärast koma (Joonis 11). Elektrivoolu andmed antud seadmel puuduvad ja nad ei ole kohustuslikud katse analüüsis vaid soovituslikud. Selleks, et leida katse kestust, mis on vajalik analüüsi jaoks, tuleb antud elektroferrogrammil võtta esimese veeru viimast väärtust ja lahutada esimene väärtus samast veerust. Selleks, et analüüsida veebirakenduses signaali väärtuseid oleks kindlasti vajalik korrutada need läbi minimaalselt 10 000-ga (korrutajat võiks määrata teadlane ise) – see on vajalik selle tõttu, et veebirakendus võtab vastu ainult täisarve ja ignoreerib kõike, mis on pärast koma. Katse meetodi kirjelduse jaoks on võimalik kasutada Drug Hunter Analyzer *metafile* põhja, aga sinna on vajalik lisada kaks lisaväärtust, mis on vajalikud katseanalüüsi jaoks, aga mis puuduvad elektroferrogrammis: sagedus (*frequency*) ja maksimaalse pingeväärtus (*highVoltageValue*).

	A	B	C
1	-0.0408333333333333	-5.56802749633789	
2	-0.0341666666666667	-5.5694580078125	
3	-0.0275	-5.5694580078125	
4	-0.0208333333333333	-5.56850433349609	
5	-0.0141666666666667	-5.56659698486328	
6	-0.0075	-5.56659698486328	
7	-0.0008333333333333	-5.56802749633789	
8	0.0058333333333333	-5.5694580078125	
9	0.0125	-5.57136535644531	
10	0.0191666666666667	-5.57374954223633	

Joonis 11. Agilent seadme elektroferogrammi näite.

### 3.3.3 CE-FLU2









CE-FLU2 on kapillaarelektroforeesi seade fluorestsentsse detektoriga. Antud seade on portatiivne aga laboris kasutamiseks, kuna iga tema ümber paigaldus vajab kalibreerimist ja lisa seadistust (Joonis 12).



Joonis 12. CE-FLU2 kapillaarelektroforeesi seade. Model - Kati Muldma, Fotograaf - Lauri Mölder. Publitseeritakse TalTech, KBI, teadurite Jelena Gorbatsova ja Jekaterina Mazina-Šinkar loaga.



CE-FLU2 seadmest tulevad välja ainult txt tüüpi elektroferrogrammid, mis samal printsiibil nagu eelnevalt analüüsitud seadmetel salvestatakse seotud arvuti kettale määratud kausta sisse. Ühe kausta sees võib olla mitu erinevat eksperimendi tulemust (Joonis 13), mis on tehtud samal kuupäeval, sama lahuse ja katse meetodiga. Meetodi fail antud seadmel puudub.

<input type="checkbox"/> Name	Status	Date modified	Type	Size
 08.02.2021_PFD-1		07.03.2021 16:36	2021_PFD-1 File	142 KB
 08.02.2021_PFD-2		07.03.2021 16:40	2021_PFD-2 File	138 KB
 08.02.2021_PFD-3		07.03.2021 16:41	2021_PFD-3 File	192 KB
 08.02.2021_PFD-4		07.03.2021 16:41	2021_PFD-4 File	195 KB

Joonis 13. CE-FLU2 seadme katsete väljundfailide list.

CE-FLU2 elektroferrogrammi andmete struktuur (Joonis 14) on sarnane Drug Hunter Analyzer struktuurile, kuigi seal on vähem andmeid. TXT tüüpi katse tulemuse failist saab leida: eksperimendi algus (*Start*) ja lõpp (*Stop*); aja samm ms (*Step*), mis on ka neljanda veeru sees; laborikatse signaali väärtus (teise veeru numbrid), pinge (kolmanda veeru numbrid). Veebirakenduses katse edaspidise analüüsi jaoks on vaja parsida antud elektroferrogrammist eksperimendi algus (*Start*) ja lõpp (*Stop*), et arvutada katse pikkust, laborikatse signaali väärtus (teises veerus asuv number). Elektrivoolu väärtust antud seadmest ei tule. Eksperimendi lõppust lahutades eksperimendi algust saab kätte eksperimendi kestuse. Signaali väärtuste täpsus antud seadme puhul ei oma numbreid pärast koma, kuna andmeid tagastatakse täisarvudena. Nii nagu Agilent seadmel, saab meetodi kirjeldamiseks kasutada sama metafile struktuuri, kus on juba lisatud sagedus (*frequency*) ja maksimaalse pinge väärtus (*highVoltageValue*).

```
1 PFD-1
2 Start 08.02.2021 10:56:11
3 Stop 08.02.2021 11:49:04
4 Step 500 ms
5 Filter Check
6 Counts 7293
7 Reference 0
8 Memo
9
10
11 Data begin
12 1 8625 3239 500
13 2 8475 3239 500
14 3 8467 3238 500
15 4 8467 3239 500
16 5 8178 3239 500
17 6 8093 3239 500
18 7 8448 3239 500
19 8 8359 3238 500
20 9 8317 3238 500
21 10 8278 3239 500
```

Joonis 14. CE-FLU2 seadme elektroferogrammi näite.

### 3.3.4 CE-C4D

CE-C4D on kapillaarelektroforeesi seade juhtivusdetektoriga. Antud seade on portatiivne, mis mahub kätte sisse, kuna tema kaal on 2 kg ja suurus on 30 x 30 x 18 (Joonis 15). Antud seade oli loodud TalTech Keemia ja biotehnoloogia instituudis, tänu sellele on võimalik kiiresti tuvastada vee sees selliseid metalle nagu liitium ja magneesium. CE-C4d seade jaoks on olemas eelnevalt kirjeldatud tööluarakendus.



Joonis 15. CE-C4D kapillaarelektroforeesi seade. Model - Kati Muldma, Fotograaf - Lauri Mölder. Publitseeritakse TalTech, KBI, teaduri Jelena Gorbatsova loaga.

Iga katse kohta luuakse eraldi kaust, kuhu lisatakse 4 faili (Joonis 16):

- experiment.json - JSON kujul salvestatud katse andmed, mida on võimalik saata veebirakenduse API kaudu analüüsi rakendusse;
- experiment\_image.png – katse jooksul mõõdetud andmed graafikuna;
- human\_readable.txt – katseandmete esitamise viis, mis on mõeldud inimese jaoks;
- ja raw.txt – algandmed, mis kasutaja saab kasutada teistes süsteemides edasiseks analüüsiks, võrdlemiseks või mingiks muuks tegevuseks.

<input type="checkbox"/>	Name	Status	Date modified	Type	Size
<input type="checkbox"/>	experiment.json		07.03.2021 16:59	JSON File	363 KB
<input checked="" type="checkbox"/>	experiment_image.png		07.03.2021 16:59	PNG File	34 KB
<input type="checkbox"/>	human_readable.txt		07.03.2021 16:59	Text Document	1 KB
<input checked="" type="checkbox"/>	raw.txt		07.03.2021 16:59	Text Document	63 KB

Joonis 16. CE-C4D seadme katsete väljundfailide list.

Antud seade juba omab veebirakendusse saatmise funktsionaalsust, aga see töötab ainult internetiühenduse olemasolul ja selle seadme jaoks on vaja luua taas-edastamist, mis

tähendab, et lõputöö raames loodav rakendus, peab oskama leida õiget tüüpi faili kaustast ja postitama selle veebirakendusse. Iga katsega edastatakse ka informatsioon, kelle konto all oli tehtud katse. See on tähtis, kuna igal teadlasel on oma loodud eksperimentide list. Analüüsi käigus selgus ka selline aspekt, et edastades eksperimendi faili veebirakendusse töölauarakendus ei edasta loodud katsefaili nime.

### 3.4 Lahendusviiside võrdlus

Antud peatükis võrreldatakse alternatiivseid lahendusviise programmeerimiskeele, mille põhjal on tehtud programmeerimiskeele valiku otsus. Antud töö põhieesmärgiks oli kiiresti teostada rakendus, mille arendamise ja kasutamisega saab hakkama erineva kogemusega inimene, mis oleks lihtsasti hallatav ja kus lihtsa paranduse jaoks pole vajalik arendaja abi. Eelnimetatud põhjuste tõttu lisati võrdluse (Tabel 1) ühe valikuna Python. Teisena oli autori poolt valitud Java, kuna see on praegu samal populaarsuse tasemel, nagu Python. [9]

Tabel 1. Programmeerimiskeelte võrdlus.

	Python	Java
Populaarsus	Väga populaarne	Väga populaarne
Süntaks	Lihtne õppida ja kasutada	Kompleksne sisaldab õppimiskõverat
Jõudlus	Java-st aeglasem erinevates rakendustes	Suhteliselt kiire
Platvormiülene	Jah	Jah, tänu <i>JVM</i> -ile
Masinõppe raamatukogud	Jah	Jah
Kompileerimine	Ei	Jah
Fragmenteerimine	2 peamist versioone	Palju

Python on objekt-orienteeritud skriptimise keel, mis on praegu laialt kasutusel masinõppe (machine learning) ja andmekäve (data-science) valdkondades. Java on objekt-orienteeritud programmeerimiskeel, mida väga laialt kasutatakse, aga mis nõuab arendajalt rohkem kogemust ja vajaliku kogemuse puudumisel võib rakenduse loomine võtta palju rohkem aega.

Analüüsi käigus tuli välja, et teadlastel juba on olemas kogemus Python rakendustega ja Python on arvutitele juba installeeritud. Samas arvestades olemasolevat aega rakenduse loomiseks ja hinnates oma Java teadmisi, langes valik programmeerimiskeele Python kasuks.

### **3.5 Nõuded loodavale süsteemile**

Järgnevalt tuuakse välja nõuded rakenduse kohta, mis selgusid teostatud analüüsi käigus. Need on kooskõlastatud rakenduse lõppkasutajaga konsulteerides.

#### **3.5.1 Nõuded loodavale rakendusele**

- Rakenduses peab olema realiseeritud autentimine veebirakendusse
- Rakendus peab oskama lugeda vajalikke andmeid .txt, .csv, .xml ja .json formaadis failidest (vajalik seadmete ja nende failides oleva andmete kaardistamine on teostatud analüüsi käigus) (Lisa 4)
- Rakendus peab võimaldama kasutajal valida: mis tüüpi seadme katsetega tegu on ja mis on eksperimendi tüüp
- Rakendus peab võimaldama kasutajal sisestada korrutajat, mis on vajalik signaali väärtuste korrektseks edastamiseks
- Elektrivoolu andmeid tuleb edastada täisarvudena ja nende puhul numbreid pärast koma pole vaja arvestada
- Rakenduses peab olema realiseeritud eksperimentide postitamine veebirakendusse ja taas-edastamise funktsionaalsus

- Iga veebirakendusse postitatud katsel peab kindlasti olema: eksperimendi tüüp, eksperimendi staatus (vaikimisi iga uue katse puhul peab olema 'new') ja eksperimendi nimi (ideaalis: katse faili nimi või kombinatsioon: katse kausta ja faili nimi)
- Vaja täiendada ja korrastada kasutaja jaoks algandmete *metafile* – lisanõue, mis ei ole rakenduse osa, aga vajalik eduka töö jaoks.

## 4 Arenduse tulemus

Neljas lõputöö peatükk keskendub tehtud töö kirjeldusele ja saadud tulemustele. Siin on kirjeldatud kasutatud metoodikat, loodud rakendust ja tema töövoogu. Valideeritakse analüüsi käigus püstitatud nõuete täitmist.

### 4.1 Kasutatud metoodika

Antud peatükis on lühidalt kirjeldatud projektis kasutatud tehnoloogiaid ja tarkvara, et saada parem ülevaade tehtud tööst.

#### 4.1.1 Microsoft Visual Studio

Tegu on Microsofti poolt loodud arenduskeskkonnaga, mis sobib peale teiste keelte ka Python projektide kirjutamiseks. Microsoft Visual Studioli on olemas nii tekstiredaktor, kui ka *Debugger*, seal on mugav failihaldus ja testimise käivitamine, ning võimalik kasutada versioonihaldust GitLab.

#### 4.1.2 Versioonihaldus GitLab

Kuna loodud rakendus on osa eelnevalt alustatud projektist, siis versioonihalduseks oli juba ette nähtud TalTech-i GitLab. See on versioonihalduse keskkond, kus on võimalik säilitada varasemaid versioone oma loodud koodist, luua uusi projekte, töötada projekti kallal kogu meeskonnaga, pidada nimekirja tehtud ja tegemata ülesannetest, luua Wiki dokumentatsiooni ja teha palju muud.

Arenduse käigus loodud failid ja kood on <https://gitlab.cs.ttu.ee/water-analyzer/scripts> repositooriumis ExperimentParser projektis. Repositooriumi omanik on diplomitöö juhendaja Evelin Halling.

Kuna töö käigus oli kasutatud Visual Studio, siis oli lisatud ka GitLab Extension, mis aitas töötada versioonihaldusega otse arenduskeskkonnast.

### 4.1.3 Python

Kuna analüüsi tulemusena sai programmeerimiskeele valik tehtud Python kasuks, siis rakendus oli kirjutatud antud keeles. Python versioon, mis oli kasutatud projekti raames on 3.9.2. Rakenduse kirjutamisel oli kasutusel ka Python stiili juhendit - PEP8. Selleks, et kontrollida koodi PEP8 standardi vastu, oli vaja juurde installida Pylint. [10] [11] [12]

### 4.1.4 File-based andmesüsteem

Kuna tehtud rakenduse sisendina on erinevates formaatides failid koos erinevate andmete struktuuridega ja edaspidi neid salvestatakse ka failidena JSON formaadis, siis selline andmesüsteem nimetatakse *file-based*.

### 4.1.5 JSON

JSON on andmevahetuse formaat. JSON-is kasutatakse võtme-väärtuse paare, kus ühele võtmele määratakse mingi väärtus või väärtuste jada. Rakendus saadab JSON formaadis infot REST API-le, mis võtmete põhjal võtab välja vajaliku info ning salvestab selle vastavatesse andmebaasi tabelitesse. Samuti katse meetodi andmed kirjeldatakse *metafile* sees, mis on JSON formaadis.

## 4.2 Loodud rakendus

Antud töö raames oli loodud konsoolirakendus (Joonis 17), mis nõuab sisendit kasutajalt. Rakenduse edukaks tööks on vajalik: arvutisse eelinstalleeritud Python, veebirakenduses eelregistreeritud konto, tehtud katsete failid – elektroferrogrammid, eeltäidetud *metafile*, kus on kirjeldatud meetod, meetodis osalenud analüütide kirjeldused ja kontsentratsioonid või juba loodud JSON formaadis eksperimendi failid, mis on vaja postitada veebirakendusse.



```
C:\Users\anna.linskaja\AppData\Local\Programs\Python\Launcher\py.exe
Collecting scripts data from C:\Users\anna.linskaja\OneDrive - HELMES A S\Documents\Isikliik\SKRIPTI KATSEID\AGILENT:
Username: anlins
Password:
logged in using 'anlins'
Please choose device type:
1 - CE-FLU1 - Drug Hunter
2 - CE-FLU2
3 - AGILENT
4 - CEC4D (collect json data)
any key to just post jsons to the bakend
please choose and press Enter: 3
Experiment measuringPoint multiplier for 4-Agilent: 10
Experiment type for 4-Agilent:
1 - regular (default)
2 - scientist
---
any key will choose 'regular'
please choose and press Enter: 1
parsing agilent_metatype.json:
parsing 10UG.CSV:
Experiment '4-Agilent_10UG.CSV' is parsed!
parsing 15UG.CSV:
Experiment '4-Agilent_15UG.CSV' is parsed!
parsing 3UG.CSV:
Experiment '4-Agilent_3UG.CSV' is parsed!
Experiment '4-Agilent_complete' is parsed!
posting all experiments from folder 'C:\Users\anna.linskaja\OneDrive - HELMES A S\Documents\Isikliik\SKRIPTI KATSEID\AGILENT\Out'.
Experiment '4-Agilent_10UG.CSV.json' is posted!
Experiment '4-Agilent_15UG.CSV.json' is posted!
Experiment '4-Agilent_3UG.CSV.json' is posted!
Experiment '4-Agilent_complete.json' is posted!
Script execution is completed
```

Joonis 17. Loodud konsoolirakendus.

Rakenduse arendamisel on kasutatud objekt-orienteeritud programmeerimise printsiipe abstraktsiooni elementidega. Staatilise koodi analüüsi raames lisati Pylint, mis lahendas korraka kaks probleemi: vastamine koodi stiilile, mis on kirjeldatud standardis PEP8 ja vigade leidmine, mida teiste programmeerimiskeelte puhul aitab leida kompilaator. Rakendusel on olemas automaattestid, mis on lisatud selleks, et koodi muutmisel andmete töötlus poleks rikutud.

#### 4.2.1 Koostöö veebirakendusega

Selleks, et veebirakendusse eksperimente sisestada on kõigepealt vaja ennast autentida. Programmi jooksutamisel küsitakse kohe alguses veebirakenduse konto kasutajanime ja salasõna, mida kasutatakse JWT-iga autoriseerimiseks. Veebirakendusega koostööks oli loodud eraldi `class SrvClient()`, mis realiseerib objekti, mida kasutatakse rakenduses sisselogimise ja andmete postitamise funktsionaalsuse jaoks. Pärast õnnestunud autentimist, salvestatakse objektis `token_id`, mille veebirakendus tagastab ja mida edaspidi kasutatakse turvaliseks andmete sisestamiseks. Veebirakendusse mitte sisse loginud kasutaja saab kasutada sisendandmete töötluse funktsionaalsust, aga tal ei õnnestu neid saata.

#### 4.2.2 Sisendandmete töötlus

Rakenduse arendamise oodatav tulemus oli selline, et kaust rakendusega pannakse sama kausta sisse, kus on olemas kaustad ühe ja sama seadme eksperimentidega. Seadme

tuvastamiseks peab kasutaja konsolis valima, millise seame katsed ta soovib töödelda ja saata. Kui seadme tüüpi pole valitud, siis teostatakse ainult enne loodud JSON formaadis failide postitamine, nende olemasolul '\_Out' kaustas.

Juhul, kui kapillaarelektroforeesi seade on edukalt valitud, käib rakendus läbi iga kausta (v.a. rakenduse kaust), otsib seadme tüübi järgi vastavates formaatides faile ja loeb nendest määratud andmeid. Andmed salvestatakse eksperimendi sissekande mudeli järgi JSON formaadis faili, mis omakorda salvestatakse '\_Out' kausta järgnevaks postitamiseks veebirakendusse. Lisaks küsitakse kasutajalt iga töödeldava kausta puhul eksperimendi tüüpi ja korrutaja väärtust.

Selleks, et rakenduse kood oleks mugavamalt loetav kasutati abstraktsiooni, mis antud rakenduse puhul tähendab seda, et iga seadme failide parser kujutab ennast objekti, mis omab välist meetodit `parse()`. Valides konsolis seadme tüübi, valitakse ka objekt - parser, mis hakkab andmetega edasi töötama. Sel viisil on seadmespetsiifilise parseri realiseerimine peidetud meetodist, mida kutsub välja `parent (ema)` meetod – antud rakenduses `iterate_folders()`. Abstraktsioon aitab eemaldada parserite sõltuvust teineteisest, mis tähendab seda, et ühe parseri koodi muudatus ei mõjuta teise parseri tööd. Lisaks hõlbustab selline lahendusviis programmi laiendamist tulevikus (näiteks uue seade lisamine).

### 4.3 Tegevusvoog

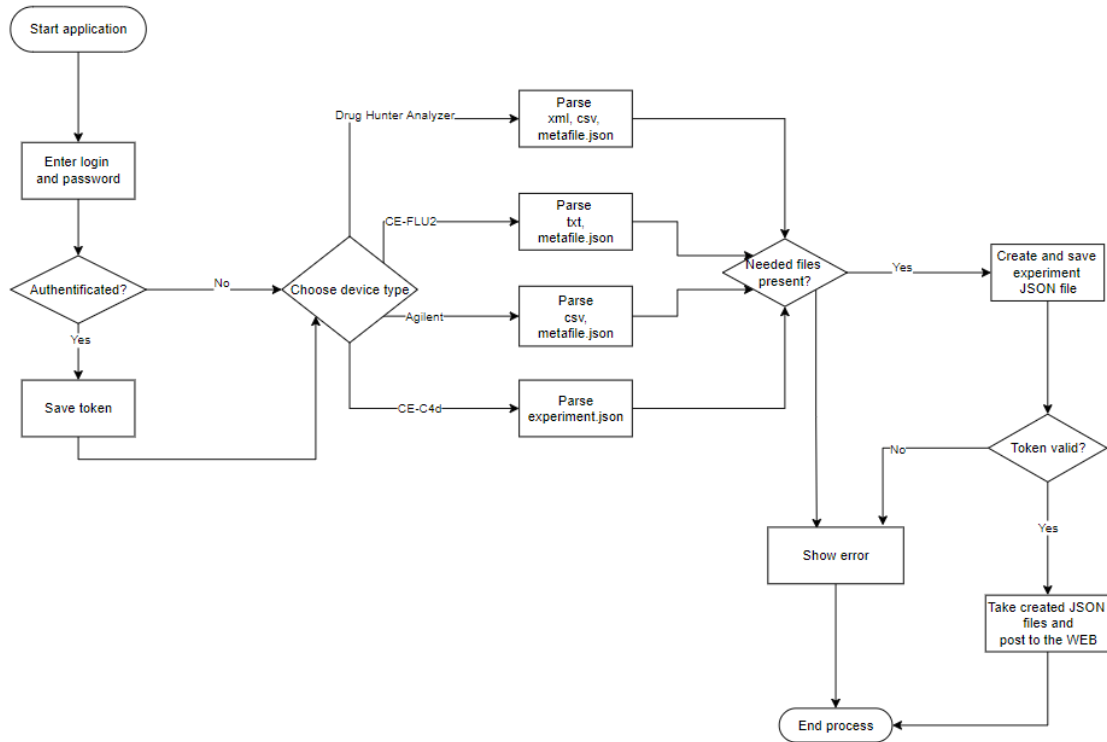
Selleks, et kasutada rakendust on vaja arvutisse installeerida Python ja kuna rakendus kasutab ka `request` teeki, siis tuleb see eraldi juurde installeerida. Viimasena nimetatud tegevuse jaoks tuleb sisestada käsk `pip3 install request` käsureal. Järgmisena on vaja täita *metafile* olemasoleva malli järgi JSON objektide kujul. Metafaili täitmisel tuleks lihtsalt olemasolevate andmete asemel uue eksperimendi andmed kirjutada, järgides sama formaati ning vajaduse korral lisades/kustutades uusi objekte. Näiteks, kui eksperimendis on vähem analüüte, kui metafaili mallina kasutatavas eksperimendis, siis tuleks lihtsalt analüütide listist analüüdi objekte eemaldada. Metafaili täitmisel on kindlasti vajalik kasutada redaktorit, mis oskab kuvada JSON-i "ilusalt" kujul (näiteks Notepad++).

Järgmisena kopeeritakse konsoolirakenduse kaust 'experiments\_parser' samasse asukohta, kus asub kaust sisendfailidega. Käivitades konsoolirakendust, palutakse

kasutajal sisse logida veebirakenduse oma kasutajanime ja parooliga. See samm on vajalik edaspidiseks turvaliseks eksperimendi andmete postitamiseks. Sisse-logimise ebaõnnestumisest või õnnestumisest annab rakendus kasutajale teada. Järgmise sammuna küsitakse kasutajalt mis tüüpi seadme andmetega tegu on, mida soovitakse töödelda, või kas on vaja lihtsalt postitada juba olemasolevad JSON vormingus eksperimente e. andmeid taas-saata. Viimati nimetatud sammu saab teha vajutades mistahes klahvi, tüüpi sisestamiseks aga valida numbrid üks kuni neli. Pärast seadme valikut rakendus küsib korrutajat ja katse tüüpi, kusjuures vaikimisi on seadistatud „regulaar“ tüüp.

Vastavalt kasutaja poolt valitud seadme tüübile hakkab rakendus otsima ettemääratud formaatides faile ja nendes määratud väärtusi. Kui süsteem ei leia vajalikku faili või andmestikku, siis väljastab ta vea ja ei loo JSON-i. Rakendus käib läbi iga olemasoleva elektroferrogrammi või kausta nendega. Kui kogu info on kokku korjatud, siis ta loob '\_Out' kausta, kuhu ta järjest lisab vastavalt veebirakenduse REST API päringu mudelile JSON faile. Neid faile luuakse sama palju, kui oli elektroferrogramme ja lisatakse ka üks koond fail, kus on kombineeritud andmed kõikidest töödeldud elektroferrogrammidest. Iga loodud JSON fail nimetatakse järgmise reegli järgi: kausta nimi + '\_' + elektroferrogrammi faili nimi; koondfaili nimi on: kausta nimi + '\_complete'.

Viimase sammuna loodud eksperimendi andmed võetakse '\_Out' kaustast ja postitakse REST API otspunkti */api/internal/experiments* kaudu ainete tuvastamise rakendusse, luues seal uusi sissekandeid (Joonis 18). Rakendus annab kasutajale konsoolis teada, kas andmed edastati edukalt või mitte.



Joonis 18. Rakenduse töövoog.

Iga rakenduse poolt postitatud eksperiment on 'new' staatusega ja kajastub sisse logitud kasutaja konto all 'My experiments' lehel. Postitatud andmeid saab kontrollida: avades signaali graafiku ja vaadates meetodi kirjeldust.

## **5 Testimine**

Tulemuste ja arenduse valideerimisel oli kasutusel mitu valideerimismeetodit – manuaalne testimine, kasutajakogemuse testimine ja automaattestimine.

### **5.1 Rakenduse manuaalne testimine**

Manuaalne testimine toimus testseadmega, mis edastati teadlaste poolt analüüsi faasis. Tudengile loodi testkonto veebirakenduses, millega oli võimalik kontrollida, kas rakenduse poolt töödeldud andmed olid postitatud ja kas nad olid korrektsed või mitte. Korrektsust kontrolliti joonistades Excel-is signaali graafikut ja võrreldes seda veebirakenduses analüüsi tulemusel saadud graafikuga.

### **5.2 Kasutajakogemuse testimine**

Koodist aegajalt töövalmis versioone saavutades toimusid online koosolekud kaasjuhendaja Jelena Gorbatšova ja teadlasega Jekaterina Mazina-Šinkar, kus toimus koos rakenduse testimine. Teadlased kontrollisid postitatud eksperimente veebirakenduses ja võrdlesid neid sisendandmetega. Vastavalt nende tagasisidele tehti muudatused koodis. Ühel hetkel tuli testimise käigus välja, et rakendus postitab eksperimente mitte ühe kaupa vaid lisades iga kord järgmise juurde. Antud viga tõi välja uue nõude, et eksperimendi postitamisel veebirakendusse peab olema loodud ka koond eksperimendi kanne, sest see lihtsustab mitme eksperimentide võrdlemist.

### **5.3 Automaattestimine**

Rakenduse kood on kaetud testidega, mis käivitavad rakenduse ja võrdlevad, et loodud JSON-id vastavad oodatavate JSON-itele. Testid on kirjutatud iga kapillaarelektroforeesi seadme jaoks.

## 6 Kokkuvõte

Antud töö eesmärgiks oli luua töötav konsoolirakendus, mis on võimeline erinevatest analüüsiseadmetest väljastatud elektroferrogrammidest looma JSON formaadis katse faile ja mis omab ainete tuvastamise veebirakendusse postitamise funktsionaalsust - luues sinna uue eksperimendi sissekande.

Püstitatud eesmärgiga ma sain hakkama ja kõik kokkulepitud nõuded olid kaetud – teadlased jäid rahule teostatud tööga ja võtsid loodud rakenduse kasutusele, kuna see aitab hoida andmeid ühes kohas ja lihtsustab edaspidi nende avaldamist. Samas see ei tähenda, et teostatud rakendus on ideaalne ning ei vaja parandamist ja täiendamist, sest kunagi pole võimalik saavutada ideaali ja mõned lisanõuded või vead tulevad välja ainult tiheda testimise ja kasutamise käigus.

Hinnates, mis oli programmi loomisel kõige raskem, siis ma saan öelda, et see on analüüsi osa ja erinevate failide töötlemine. Oli vaja aru saada, mis tüüpi seadmest, mis andmed väljuvad ja kas need on tähtsad edaspidise eksperimendi analüüsi jaoks või mitte. Kuna rakenduse arendamine oli seotud kapillaarelektroforeesi teemaga, siis lõputöö käigus sain palju uusi teadmisi analüütilisest keemiast ja sellest, miks need seadmed vajalikud on.

## Kasutatud kirjandus

- [1] „Wikipedia,“ 2020. [Võrgumaterjal]. Available: <https://et.wikipedia.org/wiki/Silumine>. [Kasutatud 30 04 2021].
- [2] C. Rohrer, „Nielsen Norman Group,“ 12 10 2014. [Võrgumaterjal]. Available: <https://www.nngroup.com/articles/which-ux-research-methods/>. [Kasutatud 04 30 2021].
- [3] „Wikipedia,“ [Võrgumaterjal]. Available: <https://et.wikipedia.org/wiki/Kapillaarelektroforees>. [Kasutatud 29 04 2021].
- [4] A. Manz, N. Pamme ja D. Iossifidis, Bioanalytical chemistry, London: Imperial College Press, 2004, pp. 69-75.
- [5] J. Gorbatošova, Kapillaarelektroforeetiliste katsete digitaliseerimise võimalused, Research work, 2019, pp. 8-22.
- [6] A. Loopalu, Töölauarakendus kapillaarelektroforeesi teostamiseks, Research work, 2018, pp. 11-46.
- [7] M. Ivanova ja R. Bondarev, Kapillaarelektroforeesi andmetöötluse töölaua-ja veebirakendus, Research work, 2020, pp. 23-39.
- [8] „Agilent,“ [Võrgumaterjal]. Available: <https://www.agilent.com/en/product/capillary-electrophoresis-ce-ms/ce-ce-ms-systems/7100-ce-system>. [Kasutatud 30 04 2021].
- [9] „PYPL PopularitY of Programming Language,“ 04 2021. [Võrgumaterjal]. Available: <https://pypl.github.io/PYPL.html>. [Kasutatud 29 04 2021].
- [10] „Python,“ [Võrgumaterjal]. Available: <https://docs.python.org/3.9/>. [Kasutatud 29 04 2021].
- [11] „PEP 8 -- Style Guide for Python Code,“ 2001.
- [12] „Pylint User Manual,“ 2021.
- [13] „Drug Hunter,“ [Võrgumaterjal]. Available: <https://www.drughunter.eu/overview>. [Kasutatud 30 04 2021].
- [14] A. Sweigart, „Working with CSV files and JSON data,“ 2020. [Võrgumaterjal]. Available: <https://automatetheboringstuff.com/2e/chapter16/>. [Kasutatud 30 04 2021].
- [15] A. Gupta, „How to read most commonly used file formats in Data Science (using Python)?,“ 2017.

## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Mina, Anna Linskaja

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Analüüsiseadmete integreerimine ainete tuvastamise rakendusega“, mille juhendaja on Evelin Halling
  - 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.

17.05.2021

---

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



## Lisa 2 – Eksperimendi loomise mudel, mis on kasutatud `/api/internal/experiments` API sees

```
Experiment {  
    analytesOfInterests (Array[AnalytesOfInterest], optional),  
    experimentCommentary (string, optional),  
    experimentDuration (number),  
    experimentLocation (ExperimentLocation, optional),  
    experimentName (string, optional),  
    experimentResults (Array[ExperimentResults], optional),  
    experimentStatus (string, optional),  
    experimentTime (string),  
    experimentType (string),  
    id (integer, optional),  
    login (User, optional),  
    method (Method, optional)  
}
```

```
AnalytesOfInterest {  
    analyte (Analyte, optional),  
    analyteConcentration (number),  
    concentrationUnit (string),  
    experiment (Experiment, optional),  
    id (integer, optional)  
}
```

```
ExperimentLocation {  
    experiments (Array[Experiment], optional),  
    id (integer, optional),  
}
```

```

        latitude (number),
        longitude (number)
    }
ExperimentResults {
    currentValue (integer, optional),
    experiment (Experiment, optional),
    frequency (integer, optional),
    id (integer, optional),
    measuringPoint (integer, optional)
}
User {
    activated (boolean),
    email (string, optional),
    firstName (string, optional),
    id (integer, optional),
    imageUrl (string, optional),
    langKey (string, optional),
    lastName (string, optional),
    login (string),
    resetDate (string, optional)
}
Method {
    analytes (Array[Analyte], optional),
    bgeCompositionOfInterests (Array[BgeCompositionOfInterest], optional),
    calibrationLines (Array[CalibrationLine], optional),
    capillaryType (CapillaryType, optional),
    description (string, optional),
    experimentProcessParams (ExperimentProcessParams, optional),
    experiments (Array[Experiment], optional),
    frequency (integer),
    highVoltageValue (number),

```

```

    id (integer, optional),
    injectionAmount (integer),
    injectionTime (integer, optional),
    injectionType (string, optional),
    injectionUnit (string),
    matrixList (MatrixList, optional),
    methodExecutionPlans (Array[MethodExecutionPlan], optional),
    methodName (string)
}

Analyte {
    analyteGroup (string, optional),
    analyteName (string),
    analytesOfInterests (Array[AnalytesOfInterest], optional),
    calibrationLines (Array[CalibrationLine], optional),
    experimentPeaks (Array[ExperimentPeaks], optional),
    id (integer, optional),
    molarMass (number, optional)
}

BgeCompositionOfInterest {
    bgeAmount (number, optional),
    bgeComposition (BgeComposition, optional),
    bgeUnit (string, optional),
    id (integer, optional),
    method (Method, optional)
}

CalibrationLine {
    analyte (Analyte, optional),
    calibrationLineData (Array[CalibrationLineData], optional),
    concentrationLimitInfo (string, optional),
    concentrationLimitMax (number, optional),
    concentrationLimitMin (number, optional),

```

```

        concentrationMax (number, optional),
        concentrationMin (number, optional),
        concentrationUnit (string, optional),
        id (integer, optional),
        intercept (number, optional),
        lodSlope (number, optional),
        lodUCL (number, optional),
        method (Method, optional),
        slope (number, optional),
        timeMax (integer, optional),
        timeMin (integer, optional)
    }
    CapillaryType {
        capillaryLength (number),
        capillaryName (string),
        effectiveLength (number),
        id (integer, optional),
        innerDiameter (number),
        methods (Array[Method], optional),
        outerDiameter (number)
    }
    ExperimentProcessParams {
        analyzeSt (integer, optional),
        batchSize (integer, optional),
        experimentAnalysis (ExperimentAnalysis, optional),
        id (integer, optional),
        methods (Array[Method], optional),
        threshold (number, optional),
        windowSize (integer, optional)
    }
    MatrixList {

```

```

        id (integer, optional),
        matrixName (string, optional),
        methods (Array[Method], optional)
    }
MethodExecutionPlan {
    checkPoint (string),
    frequency (integer),
    id (integer, optional),
    method (Method, optional)
}
ExperimentPeaks {
    active (boolean, optional),
    analyte (Analyte, optional),
    analyteConcentration (number, optional),
    analyteConcentrationUnit (string, optional),
    experimentAnalysis (ExperimentAnalysis, optional),
    id (integer, optional),
    peakArea (number, optional),
    peakEnd (integer, optional),
    peakHighest (integer, optional),
    peakNumber (integer, optional),
    peakStart (integer, optional),
    sysStatus (string, optional)
}
BgeComposition {
    backgroundElectrolyte (string, optional),
    bgeCompositionOfInterests (Array[BgeCompositionOfInterest], optional),
    id (integer, optional)
}
CalibrationLineData {
    area (number, optional),

```

```

    belowXConfidence (number, optional),
    belowYConfidence (number, optional),
    calibrationLine (CalibrationLine, optional),
    concentration (number, optional),
    deltaX (number, optional),
    deltaY (number, optional),
    id (integer, optional),
    upperXConfidence (number, optional),
    upperYConfidence (number, optional)
}

ExperimentAnalysis {
    experiment (Experiment, optional),
    experimentAnalysisData (Array[ExperimentAnalysisData], optional),
    experimentPeaks (Array[ExperimentPeaks], optional),
    experimentProcessParams (ExperimentProcessParams, optional),
    id (integer, optional),
    negativeInitialValues (boolean, optional),
    rangeFrom (integer, optional),
    rangeTo (integer, optional),
    sysStatus (string, optional)
}

ExperimentAnalysisData {
    experiment (integer, optional),
    experimentAnalysis (ExperimentAnalysis, optional),
    id (integer, optional),
    initialValue (integer),
    measuringPoint (integer),
    movingAverageSubtracted (integer, optional),
    peakIndication (integer, optional),
    smoothed (integer, optional)
}

```

## Lisa 3 – Drug Hunter Analyzer seadmest saadud XML failist katse sagedus ja pinge

```
<StepDescriptor xsi:type="AnalysisStep">
  <Substeps>
    <StepDescriptor      xsi:type="PrecisionInjectStep"      VialNumber="4"
mBarSeconds="225" PressureMax="40">
      <Substeps />
      <Translations>
        <Translation Language="Estonian">
          <TranslationItem itemName="Text" itemText="Süstimine viaalist nr 4..." />
        </Translation>
        <Translation Language="English">
          <TranslationItem itemName="Text" itemText="Injection from vial no 4..."
/>
        </Translation>
      </Translations>
    </StepDescriptor>
    <StepDescriptor xsi:type="SelectAndPunctureStep" VialNumber="5">
      <Substeps />
      <Translations>
        <Translation Language="Estonian">
          <TranslationItem itemName="Text" itemText="Kastmine viaali nr 5..." />
        </Translation>
        <Translation Language="English">
          <TranslationItem itemName="Text" itemText="Cleaning from vial no 5..." />
        </Translation>
      </Translations>
    </StepDescriptor>
```

```

    <StepDescriptor xsi:type="MeasurementStep" VialNumber="8" Voltage="25000"
CurrentLimit="120" LampVoltage="600" Frequency="210" PMTVoltage="700" ChipPump="9"
ChipVial="Right" Duration="720000">

    <Substeps />

    <Translations>

    <Translation Language="Estonian">

    <TranslationItem itemName="Text" itemText="Mõõtmise vialist nr 8..." />

    </Translation>

    <Translation Language="English">

    <TranslationItem itemName="Text" itemText="Measurement from vial no 8..."
/>

    </Translation>

    </Translations>

    </StepDescriptor>

</Substeps>

<Translations />

</StepDescriptor>

```



# Lisa 4 – Rakenduse sisendfailide võrdlus

Välj nimi väljastatud .json file-s method	Seade Falli tüüp	CE-FLU1 - Drug Hunter	CE-FLU2	AGILENT	CE-C1D - toblaan rakendusest tuleb õige JSON. Vaja saamiseks funktsionaalsust, et tagantjäreli saata veebruarikandlusse experiment.json
	CSV	XML	PFD	CSV	
capillaryType	capillaryName	methodParams: capillaryName	methodParams: capillaryName	methodParams: capillaryName	
	capillaryLength	methodParams: totalLength	methodParams: totalLength	methodParams: totalLength	
	effectiveLength	methodParams: effectiveLength	methodParams: effectiveLength	methodParams: effectiveLength	
	innerDiameter	methodParams: innerDiameter	methodParams: innerDiameter	methodParams: innerDiameter	
	outerDiameter	methodParams: outerDiameter	methodParams: outerDiameter	methodParams: outerDiameter	
	injectionAmount	methodParams: strength	ceParams: strength	ceParams: strength	
	injectionType	ceParams: type	ceParams: type	ceParams: type	
	injectionUnit	ceParams: strengthUnit	ceParams: strengthUnit	ceParams: strengthUnit	
	injectionTime	ceParams: time	ceParams: time	ceParams: time	
	matrixList:				
	matrixName	sampleMatrix	sampleMatrix	sampleMatrix	
	bgcCompositionOfInterests	bgcCompositionOfInterests	bgcCompositionOfInterests	bgcCompositionOfInterests	
	analytes:				
	analyteName	sample: compounds: compound	sample: compounds: compound	sample: compounds: compound	
	analyteGroup	sample: analyteGroup	sample: analyteGroup	sample: analyteGroup	
frequency		Frequency	Frequency	frequency	
highVoltageValue		Voltage	highVoltageValue	highVoltageValue	
experimentResults:	measuringPoint - signal				
	currentValue - voolugaigus				
experimentLocation	latitude: 59.440865				
	longitude: 24.809511				
experimentType	info: 'input' 'valiku väljast				
experimentStatus	default: väärus 'new'				
experimentName	väärus: mis koostab rakendus				
user	info: 'new' 'väljast				
	activated: true or false				
experimentAnalyses:	login				
analysesOfInterests:	tbl / list				
experimentDuration	tbl / list				
experimentTime	End' time - 'Start' time	date	Stop' time - 'Start' time	date	viimane väärtus 1. veerus - esimene väärtus 1. veerus
experimentCommentary		comments	comments	comments	
analysesOfInterests:	analyteConcentration	sample: compounds: concentration	sample: compounds: concentration	sample: compounds: concentration	
	concentrationUnit	sample: compounds: concentrationUnit	sample: compounds: concentrationUnit	sample: compounds: concentrationUnit	
	analyte	sample: compounds: compound	sample: compounds: compound	sample: compounds: compound	