

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Asko Laaniste 192863IADB

Eksperimendi disaini mikroteenus

Bakalaureusetöö

Juhendaja: Kristiina Hakk
PhD

Kaasjuhendaja: Riin Rebane
PhD

Tallinn 2023

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: Asko Laaniste

15.05.2023

Annotatsioon

Käesoleva bakalaureusetöö eesmärk on arendada mikroteenus, mis võimaldab läbi viia eksperimendi disaini planeerimist ning toorandmete pealt automaatsete tulemuste arvutamist. Mikroteenus ühildatakse juba olemasoleva tarkvara ValChrom külge, mida kasutatakse Tartu Ülikooli veebikursusel „LC-MS meetodite valideerimine“.

Tööst leiab eksperimendi disaini mikroteenus arendamise jaoks tehtud eelanalüüsi, mis koosnes intervjuust veebikursuse andjatega, osalejate küsitlusest ja kirjandusülevaatest. Seejärel esitatakse arenduse jaoks vajalikud tehnoloogilised valikud vastavalt nõuete prioriseerimise tulemustele.

Lõputöö tulemusena valmib teenus, mis on kasutatav järgnevates veebikursuse tsüklites. Veebiteenus esialgset versiooni on võimalik edasi arendada vastavalt vajadusele, kuna eksperimendi disaini võimalused on oluliselt suuremad, kui antud töö skoopi mahub.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 44 leheküljel, 7 peatükki, 15 joonist, 10 tabelit.

Abstract

Microservice for Design of Experiments

The goal of this Bachelor´ thesis was to develop a microservice for design of experiments (DoE). The service should allow user to conduct experiment planning, importing of raw data and automatic results calculation based on the provided data. This microservice will be part of ValChrom software that is used in the international web course „LC-MS method validation“ given by Chair of Analytical Chemistry of University of Tartu. The goal of the web course is to educate both chemistry students as well as practicing lab specialists in the validation of analytical methods. The developed microservice is both useful for teaching purposes and for actual use during analytical method development and validation.

The thesis contains analysis prior to development. Interviews with the web course developers were conducted, questionnaire of the web course participants to get an idea of the use of DoE among them and literature overview of the most used DoE methods. Based on the requirements and their prioritization suitable technologies were selected for development of the software.

The result of the thesis should be a useable microservice for the next iteration of the web course. The scope of the microservice is kept smaller than possible, considering the extent and capabilities of DoE methods, in order to fit in the scope of the thesis. However, the microservice is developed with the assumptions that future extension and additions would be possible.

The thesis is in Estonian and contains 44 pages of text, 7 chapters, 15 figures, 10 tables.

Lühendite ja mõistete sõnastik

analüüsimetod	analüütilises keemias kasutatav meetod huvipakkuva proovi analüüsimiseks, näiteks vedelikkromatograafia (LC) analüüsimetodi abil on võimalik määrata mõne spetsiifilise aine sisaldus proovis (putukamürki mullaproovis vms)
andmekonveier	<i>(deployment) pipeline</i> , automaatprotsessid, mille tulemusel uuendused tarkvara koodivaramus saavad test- või toodanguserverisse kasutajatele kättesaadavaks
ANOVA	<i>Analysis of Variance</i> , dispersioonanalüüs, statistiline andmeanalüüsi meetod hindamiseks rühmadevahelisi erinevusi
API	<i>Application Programming Interface</i> , rakendusliides, programmeerimist abistav vahelüli, mis lihtsustab erinevate tarkvarakomponentide omavahelist suhtlemist
BBD	<i>Box-Behnken Design</i> , eksperimendi disaini optimeerimise meetod, tuntud efektiivsuse poolest
CBA	<i>Component-Based Architecture</i> , tarkvaraarenduse lähenemisviis, mis keskendub iseseisvatele komponentidel võimaldades koodi suurt modulaarsust ja taaskasutatavust
CCD	<i>Central Composite Design</i> , eksperimendi disaini optimeerimise meetod, efektiivsem kui faktoriaaldisainid
CORS	<i>Cross-Origin Resource Sharing</i> , veebiturvalisuse kontseptsioon, mis määrab, kuidas erinevat päritolu ressursside jagamist oleks võimalik turvaliselt ning paindlikult kontrollida
CSRF	<i>Cross-Site Request Forgery</i> , veebiturvalisuse rünne, mis seisneb ohvri veebilehitseja sundimist tegema ootamatuid ja kahjulikke operatsioone ohvri volitustega
CSS	<i>Cascading Style Sheets</i> , veebilehtede visuaalse kujunduse määramiseks kasutatav arenduskeel
CSV	<i>Comma Separated Values</i> , failivorming, mida kasutatakse andmete salvestamiseks ja vahetamiseks
DDoS	<i>Distributed Denial of Service</i> , ummistusrünnak, eesmärgiks on veebiteenuse päringutetulvaga üle koormata põhjustades teenuse töös katkestusi
DoE	<i>Design of Experiments</i> , eksperimendi disain
eksperimendi disain	<i>Design of Experiments (DoE)</i> on eksperimendi planeerimise meetod, mille abil teostatakse analüüsimetodiga võimalikult

	efektiivselt eksperimente nõnda, et saadud tulemusi oleks võimalik statistiliselt korrektselt töödelda ja tõlgendada
eksperiment	(labori)katse, antud töös kasutusel analüütilise keemia terminina viitamaks laboris läbiviidavatele analüüsidele kasutades selleks spetsiaalselt ette nähtud aparatuuri, eksperiment koosneb enamasti proovi ettevalmistusest ning analüüsi teostamisest
faktor	eksperimenti disaini sisendparameeter, mille muutmine võib põhjustada analüüsimeetodi tulemusel muutuse
FD	<i>Factorial Design</i> , eksperimenti disaini üldine meetod, jaguneb mitmeks alammeetodiks, üks lihtsamaid disaine
FFD	<i>Full Factorial Design</i> , eksperimenti disaini optimeerimise meetod, üks lihtsasti arusaadava ja arvutatava struktuuriga meetod
FrFD	<i>Fractional Factorial Design</i> , eksperimenti disaini sõelumise meetod, võimaldab vähem eksperimente vähendatud tulemuste detailsuse arvelt
HTML	<i>HyperText Markup Language</i> , veebilehtede loomisel kasutatav märgendkeel
Java	objektorienteeritud programmeerimiskeel, mida kasutatakse laialdaselt mitmekülgsete rakenduste loomiseks
JavaScript	dünaamiline programmeerimiskeel, mida laialdaselt kasutatakse veebilehtede klientrakenduste arendamisel
JSX	<i>JavaScript XML</i> , JavaScripti laiend, mis võimaldab kirjutada HTML sarnast koodi JavaScripti koodi sisse, muutes koodi lihtsamini loetavaks
LC	<i>Liquid Chromatography</i> , vedelikkromatograafia, analüütilise keemia analüüsimeetod määramaks segudes erinevate komponentide sisaldust
LC-MS	<i>Liquid Chromatography Mass-Spectrometry</i> , vedelikkromatograafia-massispektromeetria meetod, analüütilise keemia analüüsimeetod määramaks segudest kõrge usaldusväärsusega erinevate komponentide sisaldust
mikroteenus	<i>microservice</i> , eraldiseisev tükk tervikrakenduses, mis on iseseisvalt uuendatav ja hallatav sõltumata suurel määral ülejäänud rakendusest; antud töö kontekstis haldab nii serveripoolseid mikroteenuseid kui ka mikroklientrakendusi
mikroklientrakendus	<i>micro frontend</i> , tükike üldisest kasutajale paistvast kuvast, mis on iseseisvalt uuendatav ja hallatav sõltumata suurel määral ülejäänud klientrakenduses osadest
MoSCoW	<i>Must have, Should have, Could have, Won't have</i> akronüüm, prioriteetide ja nõuete haldamise tehnika, mis lihtsustab arenduse ressursside haldamist

OFAT	<i>One Factor At a Time</i> , üks parameeter korraga lähenemine analüüsimeetodi robustsuse hindamiseks, kus korraga muudetakse ainult üht väärtust ning jälgitakse selle mõju analüüsitulemusele
OWASP	<i>Open Worldwide Application Security Project</i> , rahvusvaheline mitteturunduslik organisatsioon, mis on pühendunud turvalise tarkvara kirjutamise teadlikkuse levitamisele
PBD	<i>Plackett-Burman Design</i> , eksperimendi disaini sõelumise meetod, üks efektiivsemaid, kuid kasutatav ainult lihtsakoelisemaks eelfiltreerimiseks suure numbri parameetrite seast
pH	mõõtühik lahuse happelisuse või aluselise määramiseks
React	JavaScripti teek, mida kasutatakse klientrakenduste ehitamiseks
REST	<i>Representational State Transfer</i> , tarkvaraarhitektuuri paradigma, mis keskendub olekuta päringutele, klient-server eraldatusele, vahemällu salvestatavusele ning kihilistele komponentidele
RESTful API	<i>REST</i> tarkvaraarhitektuuri paradigmat kasutav API
SMTP	<i>Simple Mail Transfer Protocol</i> , standardne internetiprotokoll e-kirjade saatmiseks
TypeScript	JavaScripti laiendus, mis lisab staatilise tüübikontrolli ja objektorienteeritud lähenemise, suurendades organiseeritust ja vähendades vigade arvu koodis
ValChrom	TÜ Analüütilise Keemia õppetooli poolt arendatav tarkvara lihtsustamaks analüüsimeetodite valideerimist
VueJS	JavaScripti raamistik, mida kasutatakse klientrakenduste arendamiseks
XSS	<i>Cross-Site Scripting attack</i> , turvarünne, kus ohvri veebilehitsejasse pannakse ohtlikke skripte kasutades nt veebilehitseja sisendvorme
YAML	<i>YAML Ain't Markup Language</i> , inimloetav andmete serialiseerimise formaat, mis on laialdaselt kasutusel näiteks konfiguratsioonifailides

Sisukord

1 Sissejuhatus	12
2 Lahendatava probleemi taust.....	13
2.1 Veebikursus „LC-MS meetodi valideerimine“.....	13
2.2 ValChrom	14
2.3 Analüüsimetodite arendus ja valideerimine.....	14
2.4 Analüüsimetodi robustsuse hindamine	15
2.5 Rakenduse eesmärk	17
3 Loodava rakenduse analüüs.....	18
3.1 Nõuete kogumine.....	18
3.1.1 Kirjanduse ülevaade	18
3.1.2 Veebikursusel osalejate küsimustiku tulemused	21
3.1.3 Olemasolevate lahenduste võrdlemine	23
3.2 Nõuete prioriseerimine	25
3.3 Tagarakenduse arenduskeel	28
3.4 Tagarakenduse raamistik ja teegid	29
3.5 Klientrakenduse raamistik	31
3.6 Klientrakenduse toetavad teegid.....	32
3.7 Arendust abistavad tööriistad	33
3.8 Peamiste kuvade voodiagrammid.....	34
4 Lahenduse loomine.....	38
4.1 Tagarakenduse arendus.....	38
4.2 RESTful API tagarakenduse liides	40
4.3 Klientrakenduse arendus.....	41
4.4 Pidev integratsioon ja evitamine.....	44
4.5 Turvalisus	46
5 Loodud tarkvara.....	48
6 Tulemused ja võimalused edasiarendusteks	53
7 Kokkuvõte	55
Kasutatud kirjandus	56

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks 63

Jooniste loetelu

Joonis 1. Analüüsimeetodi lihtsustatud elukaar.	15
Joonis 2. Eksperimendi disaini vood analüüsimeetodi valideerimise kontekstis.	20
Joonis 3. Disainide kuva kasutaja tegevuste voodiagramm.	35
Joonis 4. Impordi kuva kasutaja tegevuste voodiagramm.	36
Joonis 5. Tulemuste kuva kasutaja tegevuste voodiagramm.	37
Joonis 6. Näide otspunktide defineerimisest FastAPI raamistikus.	41
Joonis 7. Näide klientrakenduses komponentideks jagamisest.	43
Joonis 8. Eksperimendi disaini teenuse paiknemine muude komponentide kontekstis.	45
Joonis 9. Disaini valimise vaade.	48
Joonis 10. Disaini parameetrite muutmise ja ekspordi vaade.	49
Joonis 11. Toorandmete impordi vaade.	49
Joonis 12. Disaini tulemuste kirjaliku kokkuvõtte vaade.	50
Joonis 13. Disaini tulemuste graafikute kuva.	51
Joonis 14. Tagarakenduse dokumentatsiooni ühe otspunkti parameetrite näide.	51
Joonis 15. Tagarakenduse dokumentatsiooni ühe otspunkti võimalike päringuvastuste vaade.	52

Tabelite loetelu

Tabel 1. Kirjandusülevaates leitud olulisemad disainid.	19
Tabel 2. Eksperimendi disainiks kasutatud tarkvarad.	22
Tabel 3. Keerulisemad osad eksperimendi disaini kasutamisel.	23
Tabel 4. Olemasolevate lahenduste võrdlus.	24
Tabel 5. Prioriseeritud funktsionaalsed nõuded.	26
Tabel 6. Tagarakenduse programmeerimiskeelte valik.	28
Tabel 7. Tagarakenduse raamistiku valimine.	30
Tabel 8. Eksperimendi disaini arvutuste jaoks sobilik teekide võrdlus.....	31
Tabel 9. Klientrakenduse raamistiku valimine.	32
Tabel 10. Versioon 1 API otspunktid.	40

1 Sissejuhatus

Antud töö eesmärk on arendada eksperimendi disaini (*Design of Experiments*, DoE) planeerimist ja arvutusi võimaldav veebiteenus Tartu Ülikooli (TÜ) Analüütilise keemia õppetooli poolt korraldatavale rahvusvahelisele veebikursusele „LC-MS meetodite valideerimine“ [1]. Veebiteenus peab olema integreeritud õppetooli poolt arendatud tarkvara ValChrom [2] külge. Teenuse eesmärk on laiendada kursuse sisu eksperimendi disaini osas [3] praktilisemate katsete ja näidetega, kui praegu võimalik on. Teenus peab võimaldama kursusel osalejatel planeerida eksperimendi disaini laborikatseid, saadud toorandmeid töödelda ning tulemusi visualiseerida, keskendudes sisule mitte keerukatele statistilistele valemitele ja tööriistade tööle saamisele, mis on hetkel praktiline paratamatus.

Antud töö autor on seotud TÜ Analüütilise keemia õppetooliga ning pakkus antud laienduse välja. Koos veebikursust korraldava meeskonnaga koostas autor vajalikud nõuded. Eksperimendi disaini teenuse kasutatavuse, veebikursusesse sobivuse ja eesmärkide prioriseerimine toimus tihedas koostöös vastava meeskonnaga (sh antud töö kaasjuhendajaga).

Töö tulemusena valmib minimaalne töötav veebiteenus, mida saab juba veebikursuse järgmises tsüklis õppetöösse integreerida (2023. aasta sügis). Algselt peab teenus võimaldama eksperimendi disaini voo läbimise planeerimisest kuni tulemuste kuvamiseni. Arendamise järgselt on plaanis veebiteenust vastavalt kursusel osalejate tagasisidele parandada ja samas teostada ka edasisiarendusi.

Lõputöö on jaotatud seitsmeks peatükiks. Töö teises peatükis antakse ülevaade veebiteenuse arenduse vajadusest ning taustast. Kolmandas peatükis kirjeldatakse nõuete kogumist, nende prioriseerimist ning nõuete põhjal arenduseks vajalike tehnoloogiate valikuid. Neljandas peatükis antakse ülevaade mikroteenuse valmimise protsessist ning esinenud väljakutsetest. Loodud tarkvarast antakse ülevaade koos vaadete kuvatõmmistega viiendas peatükis. Viimaks kirjeldatakse kuuendas peatükis püstitatud eesmärkide saavutamist ning võimalustest edasiarendusteks.

2 Lahendatava probleemi taust

Bakalaureusetöö käigus arendatava teenuse vajadus tekkis veebikursusel eksperimendi disaini osa õpetamise raskustest. Soov on keskenduda eksperimendi disaini kasutamise keemiapoolsele osale ilma, et peab palju vaeva nägema õpetamiseks vajaliku tarkvara kasutuselevõtmisega või vajama tasulist tarkvara. Eksperimendi disain on kemomeetria meetodika, mille abil saab vähendada analüütilise meetodi arendamisel ja valideerimisel vajalike laborikatsete arvu, saades samas statistiliselt usaldusväärset infot [4]. Kuigi eksperimendi disain on suurem teema, keskendub antud töö ainult analüütilise keemia meetodite valideerimisel kasutatavatele eksperimendi disaini meetoditele ja vajadustele.

Kemomeetria on analüütilise keemia haru, mis tegeleb matemaatiliste ja statistiliste võtete rakendamisega keemiliste analüüsimeetodite jaoks nii eksperimendi planeerimisel kui ka tulemuste tõlgendamisel. Vajadus kasutada statistilist andmetöötlust kasvab välja suurest muutujate hulgast, mis mõjutavad analüüsi tulemusi. Muutujad võivad olla näiteks seotud analüüsi aparatuuriga, proovide eeltötluse protseduuriga või proovide iseärasustega. [4]

Antud peatükis kirjeldatakse pikemalt probleemi tausta ning lahenduse vajadusi. Antakse ülevaade veebikursusest ja seal kasutatavast tarkvarast. Lisaks antakse ka põgus sissejuhatus analüütilise keemia analüüsimeetodite arendusse ja valideerimisse, et paremini selgitada tarkvara rolli selles protsessis. Nende sisendite põhjal püstitatakse rakenduse eesmärgid.

2.1 Veebikursus „LC-MS meetodi valideerimine“

Veebikursuse eesmärk on õpetada analüütilise keemia kromatograafiliste meetodite valideerimist nii tudengitele kui ka rutiinanalüüsi teostajatele keemialaborites. Kursust antakse alates 2017. aastast igal aastal TÜ Analüütilise Keemia õppetooli poolt nii laialdasemalt õppetooli kogemuste jagamiseks vastavas vallas kui ka õppetooli enda kompetentsi tõstmiseks. Veebikursus on inglisekeelne ja avatud kõigile soovijatele. Viimatisel sessioonil 2022/2023 õppeaastal osales 376 aktiivset inimest 104 riigist [5].

Üks peatükk kursusel on sissejuhatus eksperimendi disaini. Ent kergesti kättesaadavate ja mitte liiga mahukate abistavate tööriistade puudumisel on senini piiratud põgusa tabulaatortarkvarades teostatava harjutusega, et hoida fookus keemiapoolsel sisul. Eksperimendi disaini peatüki laiendamine saab võimalikuks antud bakalaureusetöö tulemusel valmiva tarkvara toel.

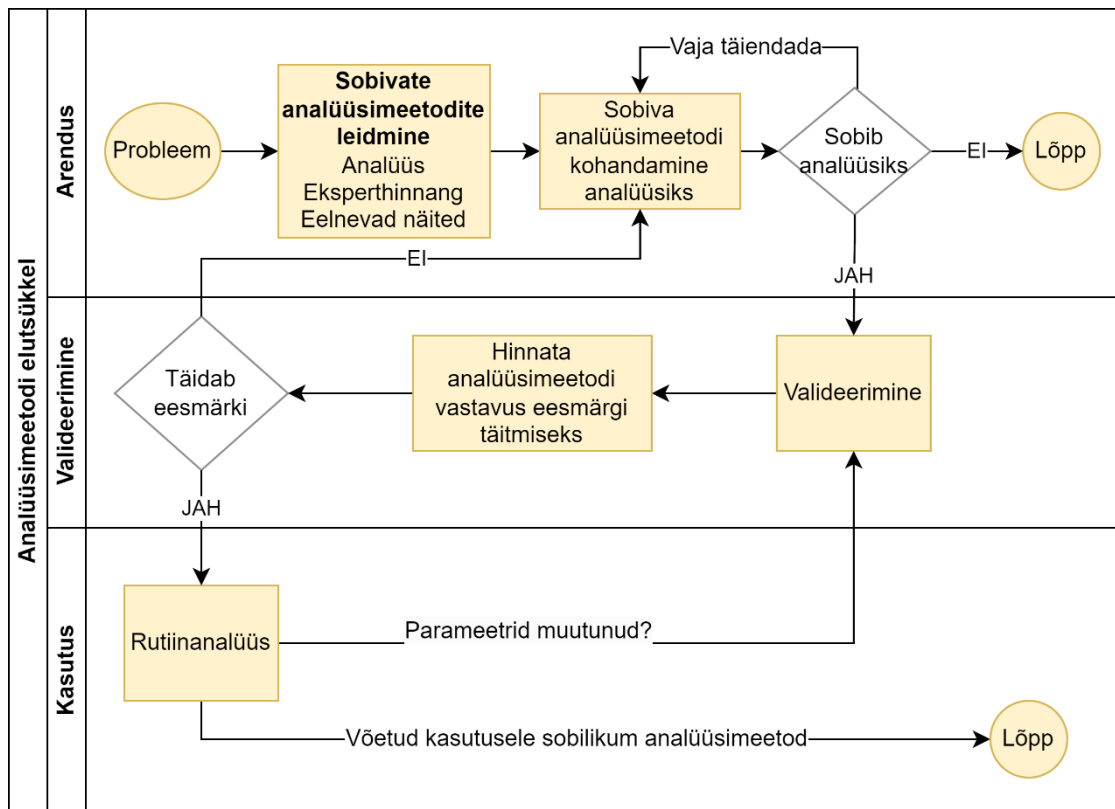
2.2 ValChrom

ValChrom on TÜ Analüütilise Keemia õppetooli poolt 2017. aastal loodud tarkvara kromatograafiliste meetodite valideerimise protsessi lihtsustamiseks [6]. Tarkvara on peamiselt kasutusel veebikursusel valideerimise automatiseerimise populariseerimiseks. Samas on tarkvara avalikult kasutada ka kõigile soovijatele veebiaadressil <https://valchrom.ut.ee>.

ValChrom tarkvara on kontseptisooni tõestus, mida arendatakse ja laiendatakse vastavalt uutele vajadustele tulenevalt veebikursuse eesmärkidest. Eelnevalt on ValChromi teemal kaitstud kaks magistritööd TÜ Tarkvaratehnika õppekaval [7], [8]. Eksperimendi disaini peatüki laiendamiseks vajalik tarkvaraline tugi lisatakse ValChrom tarkvara alla. Selleks luuakse uus teenus eraldi mikroteenusena, kuna loodav teenus on hästi piiritletav terviklik alamtükk.

2.3 Analüüsimeetodite arendus ja valideerimine

Analüüsimeetodeid kasutatakse väga erinevates eluvaldkondades, et saada täpsemalt informatsiooni keemilise koostise kohta – näiteks ravimitööstuses, keskkonnaseires, ehituskeemias, meditsiinis jne. Analüüsimeetodid usaldusväärseks rakendamiseks on vaja nii arendamises kui ka valideerimises pädevaid spetsialiste. Arenduse käigus püstitatakse eesmärgid ning seejärel valitakse sobivaimad meetodid, et need eesmärgid saavutada. Joonis 1 näitab analüüsimeetodi lihtsustatud elukaart selle arenduse, valideerimise ja kasutuse etappide läbi. [9]



Joonis 1. Analüüsimeetodi lihtsustatud elukaar.

Analüüsimeetodi valideerimise eesmärk on kindlustada, et analüüsimeetod täidab selle kasutuseks püstitatud eesmärgi. Selle käigus määratakse analüütilise meetodi parameetrite komplekt, mille muutmine oluliselt mõjutab analüüsi tulemust. Valideerimisel võib olla vaatluse all suur hulk muutujaid. Analüüsitulemust mõjutavad parameetrid tulenevad näiteks analüüsiks kasutatud instrumendist, proovi eeltötluse protseduurist, kasutatud lahustest ja materjalidest. [4]

Valideerimine omab väga tähtsat rolli ühe analüüsimeetodi elukaases. Näiteks ravimitööstuses ei saa meetodit kasutusele võtta ilma korrektse ja välise kontrollorganisatsiooni ülevaatamiseta, milleks Eestis on Raviamet. Ilma korrektsete valideerimata analüüsimeetoditeta ei ole võimalik garanteerida müüdava ravimi kvaliteeti ja vastavust riiklikele nõuetele. [10]

2.4 Analüüsimeetodi robustsuse hindamine

Analüüsimeetodi valideerimise üks komponent on selle robustsus, mis iseloomustab, kui hästi meetod täidab oma eesmärgi kui selle sisendparameetrid muutuvad selle tavapärasel

kasutamisel. Näiteks võib analüüsi käigus kasutatavate lahuste pH või kontsentratsioon olla erinevatel päevadel erinev, mis võib oluliselt mõjutada saadud tulemusi või muuta meetodi üldse kasutuskõlbmatuks. Robustsust hinnatakse peamiselt kahel meetodil: parameetri üksishaaval muutmine (OFAT, *one factor at a time*) või eksperimendi disain. [9], [11], [12]

OFAT hõlmab kõikide teiste analüüsimeetodi parameetrite väärtuste lukustamist ning ainult ühe parameetri väärtuse muutmist mõlemas suunas kesksest väärtusest. Näiteks muudetakse lahuse pH-d natukene madalamaks või kõrgemaks. Selle tulemusel saadakse selle parameetri kohta teada sobivam väärtus, mis valitakse kas uueks keskseks väärtuseks või kinnitatakse, et senine väärtus on analüüsi jaoks sobiv. Seejärel valitakse uus parameeter ning korratakse protsessi kuni kõigi eksperthinnangul väljavalitud parameetrite jaoks on leitud sobivad väärtused. [4], [13], [14]

OFAT meetod on ressursse raiskav ja saavutatakse ainult kohalikul optimaalne tulemus. Selle meetodi tulemusel eiratakse parameetrite omavahelist mõju, mis võib analüüsimeetodi tulemustes oluliselt suuremaid muudatusi esile kutsuda. Näiteks lisaks pH-le võib analüüsimeetodi tulemustele mõju omada ka lisandite kontsentratsiooni muutused ja see muutus on omakorda sõltuvuses pH muutusest. Sellised vastastikku mõjutatud parameetrid võivad koosmõjul analüüsimeetodi tulemustele omada suuremat efekti kui üksikmuutuste eraldimõõdetud summa viitab. Lisaks, OFAT puhul on parameetrite keskste väärtuste muutumisel vaja uuesti meetod osaliselt või täielikult valideerida, olles ressursi- ja ajanõudlik. [4], [13], [14]

Alternatiivne lähenemine on kasutada DoE meetodeid, mis põhinevad erinevatel statistilistel mudelitel ja analüüsimeetoditel, et saada usaldusväärsemad tulemused võimalikult vähestest eksperimentidest. See tagab tõesema arusaama kriitilistest parameetritest ning analüüsiks sobilikest parameetrite väärtustest [4]. Rakendades eksperimendi disaini juba analüütilise keemia analüüsimeetodi arenduse etapis saab usaldusväärset infot meetodi robustsuse kohta, mida saab üle kanda ka meetodi valideerimise etappi, säästes nõnda korduvate katsete tegemist. Samuti on võimalik püstitada ka analüüsiks sobilik parameetrite piirkond, mille sees parameetrite väärtuste muutumisel ei ole vajalik uuesti meetodit valideerida, vältides taaskord aja- ja ressursikulu. [13] Eksperimendi disaini kirjeldatakse lähemalt kirjanduse ülevaate peatükis 3.1.1.

2.5 Rakenduse eesmärk

Eksperimendi disaini teenusele püstitati järgmised eesmärgid:

- Võimaldab läbida vähemalt ühe eksperimendi disaini voo planeerimisest tulemusteni sõelumise ja optimeerimise etappides.
- Kasutab enim levinumaid eksperimendi disaini meetodikaid.
- Võimaldab kasutajal ilma lisapaigalduseta kohest veebiteenusena kasutamist.
- Töötleb automaatselt toorandmed tulemusteks, et kasutaja ei pea ise statistilistesse arvutustesse süüvima ega graafikute joonistamisega vaeva nägema.

Rakenduse eesmärkides piirduakse vähemate eksperimendi meetoditega (minimaalselt kaks) kui on võimalik sisse viia (kümneid erinevaid), et antud töö ajagraafikus püsiks.

Bakalaureusetöö raames arendatav versioon võetakse veebikursuse „LC-MS meetodi valideerimine“ 2023/2024 õppeaasta sessioonil kasutusele ning sealt saadud tagasiside võetakse arvesse uutes arendustes. Lisaks on plaan esialgsete arenduste pealt tagasiside koguda juba eelnevatelt veebikursuse osalistelt, et veebikursusel kasutusele võetav versioon oleks juba rohkemate kasutajate tagasiside põhjal täiustatud. Seejärel on plaanis juurde lisada ka uusi eksperimendi disaini meetodeid ja laiendada olemasolevate võimekust.

3 Loodava rakenduse analüüs

Analüüsi etapi saab jagada kolmeks suuremaks osaks. Esmalt koguti nõuete jaoks infot erinevatest allikatest. Seejärel prioriseeriti kogutud info põhjal esialgsed nõuded enne arendustegevuse algust. Viimaks uuriti, toetudes prioriseeritud nõuetele, millised tehnoloogiad on sobilikud eesmärkide saavutamiseks.

Selles järjekorras on ka järgnevad alampeatükid esitatud. Analüüsi lõpupeatükis on toodud juba arendustegevuse jooksul tekkinud voodiagrammid erinevate vajalike kuvade kohta. Voodiagrammid annavad arusaama, millised tegevused on kasutajale kättesaadavad ja mis funktsionaalsused on tarkvaras suurel määral vajalikud.

3.1 Nõuete kogumine

Funktsionaalseid nõudeid koguti kasutades autori enda varasemaid domeeniteadmisi, intervjuud veebikursuse koostajatega, veebikursuse materjale, küsitlust veebikursuse osalejatele ning kirjanduse ülevaadet eksperimendi disaini meetodikatest analüütilises keemias. Veebikursuse koostajatega on autoril juba varasemalt iganädalased koosolekud, mistõttu selle käigus anti ülevaade eksperimendi disaini teenuse arendusest ja küsiti tagasisidet.

3.1.1 Kirjanduse ülevaade

Kirjandusülevaade teostati *Web of Science* andmebaasist otsides ülevaateartikleid eksperimendi disaini ja kromatograafi teemadel ning sorteerides need enim viidatavate järgi. Tabel 1 annab ülevaate leitud disainidest, mis moodustasid esialgse valimi, mille hulgast valiti bakalaureusetöö käigus teostatavad meetodid.

Eksperimendi disaini meetodid jagatakse enamasti kaheks suuremaks grupiks: sõelumise meetodid (*screening design*) ja optimeerimise meetodid (*optimising design*). Sõelumise meetodite eesmärk on suurest hulgast sisendparameetritest leida olulisemad, mis mõjutavad eksperimendi tulemusi. Seejärel rakendatakse kõige olulisematele parameetritele (edaspidi eksperimendi disaini kontekstis faktoritele) optimeerimise

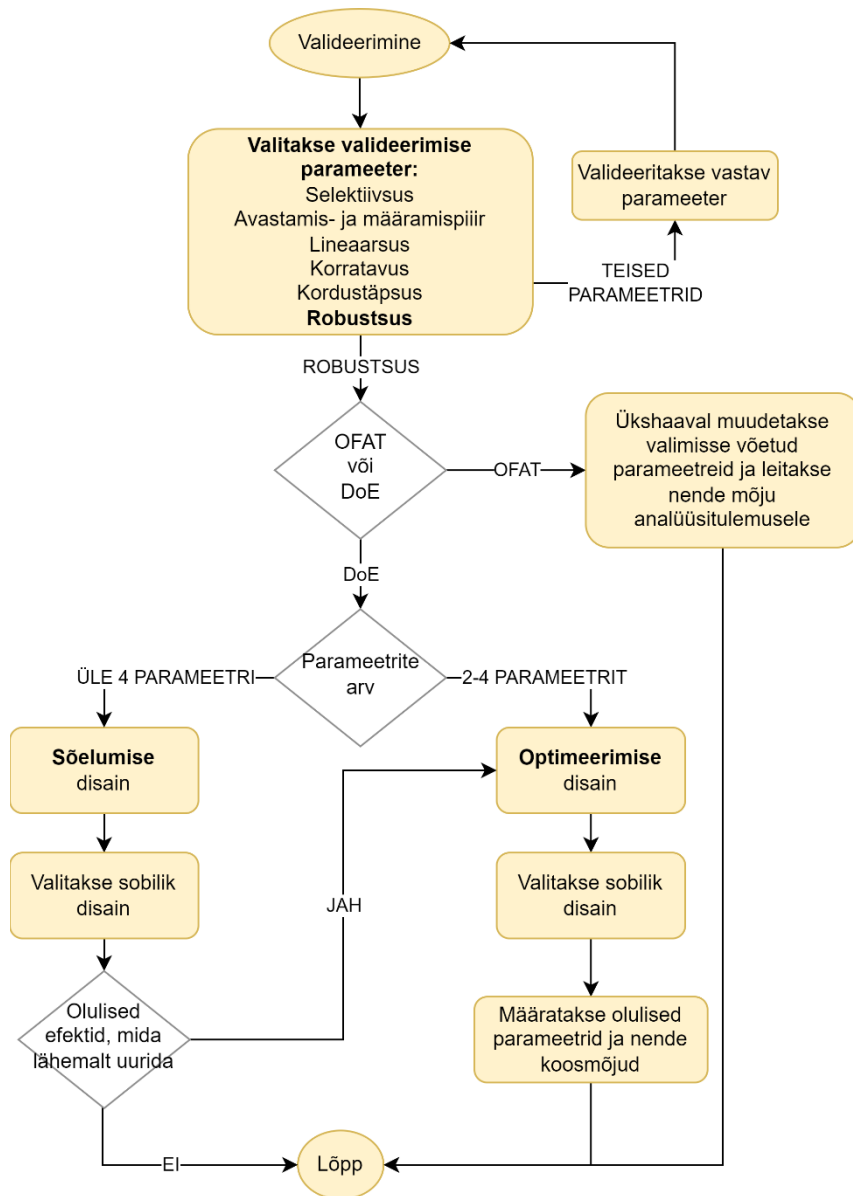
meetodid, hindamaks kui palju need mõjutavad väljundparameetri(te) tulemust. [4], [15]–[17]

Tabel 1. Kirjandusülevaates leitud olulisemad disainid.

Disain	Disaini kirjeldus	Allikad
<i>Plackett-Burman</i> (PBD)	<ul style="list-style-type: none"> ▪ sõelumise disain ▪ väga piiratud ▪ väga kokkuhoidlik 	[4], [14]–[17]
<i>Fractional Factorial</i> (FrFD)	<ul style="list-style-type: none"> ▪ sõelumise disain ▪ mitmekülgset optimeeritav ▪ kokkuhoidlik 	[4], [14]–[17]
<i>Full Factorial</i> (FD)	<ul style="list-style-type: none"> ▪ optimeerimise disain ▪ lihtsam teostada 	[4], [14]–[17]
<i>Box-Behnken</i> (BBD)	<ul style="list-style-type: none"> ▪ optimeerimise disain ▪ efektiivsem kui FD 	[4], [14]–[17]
<i>Central Composite</i> (CCD)	<ul style="list-style-type: none"> ▪ optimeerimise disain ▪ mitmekesised võimalused teostamiseks 	[4], [14]–[17]
<i>Doehlert</i>	<ul style="list-style-type: none"> ▪ optimeerimise disain 	[4], [14]–[16]
<i>D-optimal</i>	<ul style="list-style-type: none"> ▪ optimeerimise disain ▪ keerukamad algoritmid 	[4], [14]–[16]
<i>Taguchi</i>	<ul style="list-style-type: none"> ▪ optimeerimise disain 	[4], [14]
<i>Mixture</i> (Segadisainid)	<ul style="list-style-type: none"> ▪ optimeerimise disain 	[14], [15]
<i>Face-Centered</i>	<ul style="list-style-type: none"> ▪ optimeerimise disain 	[14]

Kaks sammu (sõelumine ning seejärel optimeerimine) on vajalik, kuna optimeerimise meetodikate puhul suureneb tehtavate laboriekspperimentide hulk kiirelt iga järgneva sisendparameetri kohta. Seetõttu kasutatakse sõelumise puhul eelnevalt meetodeid, mis ei võimalda nõnda täpset infot saada kui optimeerimise disainid, kuid seeläbi vajavad vähem laboriekspperimente. Olenevalt sõelumise meetodist, ei saa usaldusväärset hinnata faktorite koosmõju ning samuti võib peamisi faktorite endi kvantitatiivseid efekte olla võimatu hinnata [17]. Samas enamikel juhtudel on sõelumise meetodikad piisavad, et hinnata, millised on peamised faktorid, mida edasi vaadelda [15]. Sõelumise meetodikaid kasutatakse enamasti juhul, kui algsest on vaatluse all viis või enam analüüsimeetodi

parameetrit ja ei ole võimalik eelteadmise või eksperthinnangu alusel mõnda neist välja jätta [4], [17]. Optimeerimise meetodid on peamiselt kasutatavad kahe kuni nelja faktori puhul, olles liiga eksperimendimahukad ja ebapraktilised suurema hulga faktorite puhul [4]. Joonis 2 annab ülevaate eksperimendi disaini paigutumisest valideerimise töövoogu.



Joonis 2. Eksperimendi disaini vood analüüsimeetodi valideerimise kontekstis.

Kirjanduse ülevaate alusel on akadeemilises ringkonnas analüüsimeetodite arenduseks ja robustsuse hindamiseks enim kasutatud sõelumise meetodid *Fractional Factorial* disain (FrFD) ning *Plackett–Burman* disain (PBD) [4], [14]–[16]. Enim kasutatud optimeerimise meetodid on *Box-Behnken* disain (BBD), *Full Factorial* disain (FFD),

Central Composite disain, *Taguchi* disain, *Doehlert* disain [4], [14]–[16] ja *D-Optimal* disain [15], [16].

Eelpool nimetatud disainide põhjal teostatakse laboris mõõtmised, millest saadud andmeid on vaja edasi töödelda, et saada analüüsimeetodi headust iseloomustavad tulemused. Mõõtmisandmete põhjal koostatakse ennustusmudel, mis kirjeldab erinevate parameetrite (nt pH) mõju vaadeldavale tulemusele (näiteks analüüsimeetodi täpsus või mõõdetava signaali suurus). Ennustusmudeli põhjal saadakse teada, millised analüüsimeetodi parameetrid on olulised ja millised mitte. Mudeli analüüsimiseks kasutatakse näiteks dispersioonanalüüsi (ANOVA) [13], regressioonanalüüsi [4], põhifaktorite ja nendevaheliste interaktsioonide kvantitatiivseid hinnanguid [17] [18]. Mudeli efektide ja ennustatud tulemuste visualiseerimiseks kasutatakse näiteks kontuurdiagrammi (*countour plots*), *Pareto* diagrammi, normaaljaotuse tõenäosusdiagrammi (*normal probability plot*) [4], vastuspinna diagrammi (*response surface*) [17] ja *Box-Cox* diagrammi, lisaks paljudele teistele variantidele. [18]

Esialgse voo tõestuseks ja töö valmimise ajaraamide tõttu valiti antud töö jaoks üks populaarseim meetod nii sõelumise kui ka optimeerimise meetodite seast. Populaarsust hinnati artiklites esinemise järgi (Tabel 1) ning võrdväärsete puhul valiti esmase teostatavuse hinnangu alusel. Sõelumise disainiks valiti FrFD, kuna tegemist on ühe tuntuima meetodiga, mille eksperimentide arvu saab vastavalt vajadusele kohandada, ohverdades mõningal määral saadavate tulemuste täpsuses. Optimeerimise meetodiks valiti BBD, mis võimaldab vähem eksperimente teha kui samaväärsed CCD või FFD, mis on ühed populaarseimad meetodeid optimeerimiseks [16], [19], [20]. Näiteks kolme faktori puhul kulu FFD jaoks $3^3 = 27$ eksperimenti, kuid BBD puhul saab hakkama 13 eksperimendiga [19], [21]. Üldjuhul on kahe samaväärse meetodi puhul eelistatum vähem eksperimente teostav meetod, et laborikatsete ressursikulu piirata. Lisaks on kummagi disaini teostamine lihtsam nii arvutuslikult kui ka eksperimentide planeerimise etapis võrreldes näiteks *D-optimal* või mitmete CCD disainidega [22], [23].

3.1.2 Veebikursusel osalejate küsimustiku tulemused

2022/2023 õppeaasta veebikursuse „LC-MS meetodite valideerimine“ alguses saadeti osalejatele küsitlus seoses kursuse üldiste eesmärkide ja ka spetsiifiliselt eksperimendi disaini kohta. Eesmärgiks oli saada ülevaade, kui palju osalistest on eksperimendi disainiga kokku puutunud. Küsimustikule vastas 145 osalejat, kellest 76 olid

akadeemilise taustaga ja 71 mitteakadeemilise taustaga laboratooriumis töötavat inimest. 48 osalejat oli varem eksperimendi disainiga kokku puutunud.

Eksperimendi disaini läbiviimiseks kasutatavatest tarkvaradest (Tabel 2) polnud 68 inimest midagi kasutanud või ei kasuta eksperimendi disaini üldse. 58 inimest kasutas MS Excelit ja teisi tarkvarasid kasutati kõiki all kümne juhu. Tabulaatortarkvarade (nt MS Excel) kasutamine nõnda arvutusintensiivse kui ka keerulise statistika jaoks on problemaatiline. Vigu on raske leida ja parandada, kuna iga analüütiline keemik või ettevõtte kasutab oma varianti lahendusest, mida on töö autor ka omast kogemusest koostöös erinevate ettevõtetega näinud.

Tabel 2. Eksperimendi disainiks kasutatud tarkvarad.

Tarkvara	Vastanuid
Puudub (ei ole kasutanud eksperimendi disaini)	68
MS Excel	58
R	9
Muu tarkvara	8
SPSS	8
Minitab	7
Matlab	6
Design Expert	6
Statistica	5
Python	4
JMP	1
Modde	1

Küsimustik annab ka ülevaate eksperimendi disaini kõige keerulisematest väljakutsetest (Tabel 3). Tulemustest selgub, et eksperimentide planeerimine, eksperimendi disaini meetodites orienteerumine ja tulemuste tõlgendamine on kõik võrdväärselt keerulised. Samuti valmistab tarkvara maksumus ja/või keerukus probleeme. Kasutajad võivad oluliselt tarkvarast, mis oleks sihitud just analüütilise keemia ja kitsamalt kromatograafiaga seotud meetodite arendusele ja valideerimisele. Selline tarkvara teeb kasutajate eest ära enamuse keerukamast arvutusest ja andmete analüüsist, et keemik saab keskenduda kõige olulisematele osadele.

Tabel 3. Keerulisemad osad eksperimendi disaini kasutamisel.

Keerulisemad osad eksperimendi disaini juures	Vastanuid
Ei ole kasutanud.	62
Laboriekperimentide õige planeerimine, et tulemuste analüüs oleks pädev.	35
Millist eksperimendi disaini metoodikat kasutada.	35
Andmete tõlgendamine pärast eksperimente.	31
Arvutused on keerulised.	23
Tarkvara eksperimendi planeerimiseks ja tulemuste analüüsiks on liialt kallis.	22
Esmakordseks kasutuseks ei leia piisavalt head juhendmaterjali.	22
Tasuta tarkvara on keeruline kasutada.	14
Kaastöölised ei mõista eksperimendi disaini, mistõttu koostöös on keeruline kasutada.	12
Andmete visualiseerimine.	0
Tulemuste tõlgendamine.	0

3.1.3 Olemasolevate lahenduste võrdlemine

Eksperimendi disaini vooge abistavaid tarkvarasid on väga palju (Tabel 4) [16]. Suures osas saab need tarkvarad jagada kaheks suuremaks grupiks, mis illustreerivad selgemalt probleemi nendega alustamise osas:

1. lühikese katseajaga tasulised tarkvarad, mis on suunatud suurematele organisatsioonidele (nt ravimitööstus);
2. tasuta, kuid suure õpikõveraga ja keerulised tarkvarad.

Antud töö analüüsi käigus ei ole õnnestunud leida ideaalset varianti kolmandaks juhuks: tasuta või taskukohane ning lihtne kasutama hakata.

Tasuliste tarkvarade puhul on analüütilise keemiku jaoks kaks peamist puudust: suur kulu (tihti algavad kasutuslitsentsid 1000 EUR piirilt) ning liiga suur võimekus. Enamik tarkvarasid katavad ära suurima võimaliku juhtude arvu, mis võivad esineda eksperimendi disaini või üldiste statistiliste arvutuste vajadusel, kuid osutub suuresti ebavajalikuks analüütilise keemia analüüsimeetodite arenduse ja valideerimise kontekstis. Sobiv tarkvara on spetsialiseerunud tarkvara, mis võtab arvesse just selle valdkonna erisusi ja ei sunni suures valikus muude võimekuste vahel orienteerumist.

Tabel 4. Olemasolevate lahenduste võrdlus.

Tarkvara	Eelised	Puudused
Tabulaatotarkvara (nt MS Excel [24])	<ul style="list-style-type: none"> ▪ laialt levinud ▪ enamikel keemikutel olemas kasutamise kogemus 	<ul style="list-style-type: none"> ▪ tasuline ▪ arvutusvoog tuleb ise luua ▪ väga veaaldis
Matlab [25], Statistica [26], IBM SPSS (<i>Statistical Package for the Social Sciences</i>) [27], Minitab [28]	<ul style="list-style-type: none"> ▪ väga võimekad ▪ üldiselt head juhendid 	<ul style="list-style-type: none"> ▪ tasuline (katseperioodiga) ▪ fookus pole eksperimendi disainil ▪ pigem fookus üldisel statistikal ▪ järsk õpikõver
JMP [29], Modde [30], Fusion Pro [31], Origin [32], Statgraphics [33], Design-Expert [34]	<ul style="list-style-type: none"> ▪ väga võimekad ▪ fookus ka eksperimendi disainil ▪ Origin: võimalik R, Python, Matlab jm integratsioonid, ▪ Design-Expert: integratsioonivõimalus Pythoniga 	<ul style="list-style-type: none"> ▪ tasuline (katseperioodiga) ▪ õppimiskõver pigem järsk ▪ Origin: sõltuv R tarkvarast
R [35]	<ul style="list-style-type: none"> ▪ tasuta ▪ väga võimekas (nii arvutused kui graafiline andmete kuva) ▪ küllalt levinud ▪ palju juhendeid 	<ul style="list-style-type: none"> ▪ järsem õpikõver ▪ keerulisem süntaks tehnoloogiakaugete inimesele ▪ vajab eraldi paigaldamist igasse masinasse
Python [36]–[41]	<ul style="list-style-type: none"> ▪ tasuta ▪ väga võimekas, kui osata kasutada ▪ palju abistavaid teeke 	<ul style="list-style-type: none"> ▪ programmeerimiskeel ▪ järsk õppimiskõver ▪ osaliselt vaja vood ise üles ehitada ▪ vaja vaeva näha kohalikku masinasse tööle saamisega ▪ väga raske tehnoloogiakaugete inimesele

Teise gruppi kuuluvad tasuta tarkvarad on väga võimekad (näiteks R ja Python), kuid nõuavad märkimisväärset panust kasutajal kõigepealt nende paigaldamiseks ja kasutama

õppimiseks. Samuti peab kasutaja eksperimendi disaini töövoos sisseviimisel algselt suure ajakuluga arvestama, mida on ka töö autor ja tema kolleegid omal käel tunda saanud. Ideaalis ei pea kasutaja ise tarkvara seadistamisega vaeva nägema, vaid see on lihtsalt integreeritav olemasolevasse töövoogu ning kasutaja saab kohe tegeleda eksperimendi disaini planeerimise ja tulemuste tõlgendamisega.

3.2 Nõuete prioriseerimine

Prioriseerimise jaoks kasutati MoSCoW (akronüüm: *must-have* – *M*, *should-have* – *S*, *could-have* – *C*, *won't have* – *W*) prioriseerimise meetodit [42], kuna see oli tuttav ka veebikursuse korraldajatele, kellelt nõu küsiti. Samuti koguneb selle prioriseerimise käigus ka nõuete hulk, mida esialgu töö raames ei teostata, kuid saab võtta plaani hilisemates arendustes. MoSCoW meetod lubab piirata kõikvõimalikke potentsiaalselt teostatavaid funktsionaalsusi kriitilistest funktsionaalsustest, mida on kindlasti vaja veebikursuse eesmärkide teostamiseks.

Tabel 5 sisaldab prioriseeritud funktsionaalseid nõudeid. Kriitilisteks nõueteks on ühe sõelumise ja ühe optimeerimise disaini võimekus ning nendega seotud tegevused. Olulised funktsionaalsed nõuded sätestasid näiteks ekspordi funktsionaalsusi, ValChromiga integreerimist ja kasutajatelt tagasiside kogumise võimekust. Teenusel võiks olla ka spetsiifilisemad teadete kuvamised kasutajatele, toorandmete valideerimised ja olemasolevad funktsionaalsuste laiendamised. Prioriseerimisel leiti, et esmajärjekorras ei teostata mitmete erinevate disainide valikuid ning võimekust salvestada tulemusi (raporteerimine ega kasutajakontod).

Tabel 5. Prioriseeritud funktsionaalsed nõuded.

M – Kriitilised funktsionaalsused	S – Peaks olema
<ul style="list-style-type: none"> ▪ Kasutaja saab teenust tarbida veebirakendusena. ▪ Kasutaja saab luua ühe sõelumise eksperimendi disaini plaani. ▪ Kasutaja saab luua ühe optimeerimise eksperimendi disaini plaani. ▪ Kasutaja saab toorandmed üles laadida. ▪ Kasutaja saab näeb tootandmete pealt automaatselt arvutatud tulemusi sõelumise eksperimendi disainile. ▪ Kasutaja saab näeb tootandmete pealt automaatselt arvutatud tulemusi optimeerimise eksperimendi disainile. ▪ Kasutaja üleslaetud toorandmed ja nende põhjal arvutatud tulemused on ainult temale nähtavad. ▪ Sõelumise disainid võimaldavad hinnata faktorite kvalitatiiivset mõju tulemustele. ▪ Optimeerimise disainid võimaldavad hinnata faktorite kvantitatiivset mõju tulemustele. 	<ul style="list-style-type: none"> ▪ Kasutaja saab valida eksperimendi disaini faktorite arvu ja sisestada nende tasemed. ▪ Kasutaja saab eksperimendi disaini tulemused eksportida. ▪ Integreeritav ValChrom tarkvaraga. ▪ Optimeerimise eksperimendi disaini tulemusi on võimalik vaadelda kolmemõõtmelistel vastuspinnal graafikutel. ▪ Abijuhendid tarkvara enda vaadetes, mis aitaksid algaja kasutaja läbi eksperimendi disaini voo. ▪ Kasutaja tagasiside vorm.
C – Võiks olla	W – Ei tee
<ul style="list-style-type: none"> ▪ Kasutaja toorandmeid valideeritakse. ▪ Kasutajale kuvatakse hoiatuse- ja veateateid, kui esialgne disain ei vasta üleslaetud toorandmetele. ▪ Kasutaja saab mitmete faktorite (enam kui kolm) tulemused samale graafikule kuvada, et valida parim optimaalne tööpiirkond välja. ▪ Lisaks BBD disainile on võimalik kasutada ka FFD disaini. ▪ Kasutaja saab tulemuste graafikute kuvamise stiili muuta. 	<ul style="list-style-type: none"> ▪ Kasutaja saab valida mitme erineva sõelumise eksperimendi disaini vahel. ▪ Kasutaja saab valida mitme erineva optimeerimise disaini vahel. ▪ Kasutaja saab oma eksperimendi disaini jagada teiste kasutajatega. ▪ Tulemuste arvutamine on võimalikult selge ja kõik matemaatilised tehted on otsast lõpuni jälgitavad. ▪ Raporteerimine. ▪ Varasemate tulemuste salvestamine ja vaatamine.

Lisaks eelpool analüüsitud funktsionaalsetele nõuetele osutusid oluliseks ka mittefunktsionaalsed nõuded olemasoleva tarkvara lahendustest lähtuvalt. Olemasolev arhitektuur on mikroteenustel põhinev hajus arhitektuur: ValChrom on jagatud mitmeteks erinevateks mikroteenusteks koosnedes klientrakendusest, tagarakendusest, matemaatika

teenusest ning muudest abistavatest teenustest. Eksperimendi disaini teenus on selgelt piiritletav ülejäänud rakendusest. Samas kuulub see üldisesse ValChromi valideerimist automatiseerivasse voogu. Ühtlasi on juba olemasolevad ValChromi teenused arendatud mikroteenuste vormis. Seetõttu otsustati teenus teha mikroteenuseks olemasoleva tarkvara külge.

Mikroteenused on tarkvaraarenduse moodus, kus rakendus koosneb mitmetest väikestest ja üksteisest sõltumatutest teenustest. See võimaldab lihtsamalt hallata ja uuendada väiksemat osa rakendusest ilma vajaduseta kogu rakendust muuta. Tihti suhtlevad mikroteenused üle API (*Application Programming Interfaces*) rakendusliidese. API liides on programmeerimist abistav vahelüli, mis lihtsustab erinevate tarkvarakomponentide omavahelist suhtlemist aidates luua üksteisest sõltumatuid teenuseid ja sobib hästi kasutuseks mikroteenuste arhitektuuris. [43], [44]

Mikroteenuste loogika on algselt kasutusel olnud rohkem serveripoolsete protsesside tükeldamisel, kuid on hakanud levima ka klientrakenduste tükeldamisel mikroklientrakendusteks (*micro frontend*). Mikroklientrakendused lubavad erinevaid kasutajale kuvatavaid teenuseid üksteisest eraldatult arendada, kuigi kasutajale paistavad need endiselt ühe rakenduse osana. Mikroklientrakenduste eelised ja lähenemised on väga sarnased serveripoolsetele mikroteenustele, mistõttu edaspidi antud töö kontekstis mõeldakse mikroteenuse all mõlemaid nii mikroteenuseid serveri pool kui ka mikroklientrakendusi kasutajakuva poolel. [45]

Eelnevast lähtuvalt on arendataval veebiteenusel järgnevad mittefunktsionaalsed nõuded:

- Eraldi klient- ja tagarakendus, võimaldades rakenduse teistest komponentidest sõltumatut uuendamist.
- Rakenduse konteinerina paigaldamine test- ja toodanguserverile olemasoleva andmekonveieri (*deployment pipeline*) abil.
- API otspunktid on dokumenteeritud hõlbustamiseks lihtsamat kasutust tulevikuarendustes.

Eksperimendi disaini tagarakenduse eesmärk on peamiselt statistiliste arvutuste läbiviimine. Klientrakenduse ülesandeks on eksperimendi disaini valikute võimaldamine, toorandmete import ning arvutustest saadud mudelite pealt tulemuste ja graafikute

kuvamine. Selline ülesannete piiritlemine aitab lihtsustada ka arendusprotsessi, lubades keskenduda korraga väiksemale alamosale suuremast ülesandest.

3.3 Tagarakenduse arenduskeel

Tagarakendus on veebirakenduse osa, mis vastutab andmete töötlemise ning salvestamise, äri loogika ning seotud infrastruktuuri eest [46]. Eksperimendi disaini teenuse tagarakenduse arenduskeele valikul oli üheks oluliseks kriteeriumiks võimekus teostada vajalikke statistilisi arvutusi võimalikult lihtsalt.

Tagarakenduse arenduskeele valikul lähtuti esmajärjekorras programmeerimiskeele võimekusest teostada statistilisi arvutusi ning vastavate toetavate teekide olemasolu. Joonis 8 näitab otsingu tulemusel leitud potentsiaalselt sobilikke programmeerimiskeeli, mis võiksid võimaldada antud töös vajalikke statistilisi arvutusi.

Tabel 6. Tagarakenduse programmeerimiskeelte valik.

Programmeerimiskeel	Eksperimendi disaini toetavate teekide hulk	Arenduseks mugav	Õpiköver	Kogemus
R	Palju	Pigem piiratud	Väike	Väike
Java	Vähe	Jah	Keskmine	Suur
Python	Palju	Jah	Väike	Väike

R on statistilise arvutuse ja graafika jaoks mõeldud keel ja keskkond, mis on laialdaselt kasutatav teadlaste, statistikute ja andmeanalüütikute poolt. R keelele on palju teekke eksperimendi disaini arvutuste ja visualiseerimise jaoks, sealhulgas *DOEbase* [47], *AlgDesign* [48] ja *FrF2* [49]. Lisaks võimaldab R teha ka lihtsa veebirakenduse *shiny* teegiga. Siiski on R keele võimekus piiratud ja see sobib pigem väiksemate ja kitsa fookusega arvutuste ning visualiseerimiste jaoks, keskendudes peamiselt arvutamisele. [35], [50]

Java on üldotstarbeline programmeerimiskeel, mis on tuntud oma usaldusväarsuse, tagasiühilduvuse ja turvalisuse poolest, mistõttu on tihti populaarne valik tagarakenduste ehitamisel (ka mikroteenuste) [51], [52]. Javal on ka mõned teegid statistiliste arvutuste jaoks (*IMSL* ja *Apache* ja *Commons Math* teegid), kuid nende võimekus eksperimendi

disainile ja statistilistele arvutustele on piiratud või on nad tasulised [53], [54]. Seetõttu ei ole Java statistiliste arvutuste läbiviimiseks antud töö kontekstis mugav.

Python on dünaamiliste tüüpidega arenduskeel programmeerimiskeel, millega alustamine on lihtne ka arenduskaugele inimesele [55]. Seetõttu on ta enamasti esimene valik kui vaja statistilist andmetöötlust teostada [56], [57]. Statistiliste arvutuste ja andmeteaduse jaoks kasutatavuse populaarsus on näha ka 2021. aasta *Stack Overflow* populaarseimate teekide uuringust, kus mitmed Pythoni andmeteadusega teegid (nt *numpy*, *pandas*) olid ühed populaarsemad [58]. Pythonil on lai valik teeke eksperimendi disaini läbiviimiseks ja andmetöötuseks, näiteks *pyDOE*, *scipy*, *pandas* ja *numpy* [59]–[62].

Tagarakenduse keeleks otsustati valida Python peamiselt statistilisi arvutusi ja spetsiifiliselt eksperimendi disaini toetavate teekide rohkuse tõttu. Lisaks osutusid soodustavateks põhjusteks ka keele väike õpikõver, populaarsus ja sobilikkus professionaalseks tarkvaraarenduseks.

3.4 Tagarakenduse raamistik ja teegid

Järgnevalt vaadati Pythoni raamistikke, mis toetaksid väikese rakenduse kiiret arendust. Pythonil on mitmeid raamistikke, millest vaatluse alla jäid Flask, FastAPI ja Django. Autoril on kogemusi olnud ainult vähesel määral Django raamistikuga, mis on kasutusel ValChromi tagarakenduses.

Tabel 7 näitab raamistike võrdlust, mille tulemusel valiti edasiseks arenduseks FastAPI, kuna too on enam fokuseeritud väikemate rakenduste kiireks arenduseks ning toetab lihtsalt ka tagarakenduse otspunktide dokumentatsiooni läbi OpenAPI standardi. FastAPI raamistikus on sisseehitatud tugi OpenAPI standardile, mis aitab määratleda API dokumenteerimiseks ja kasutamiseks vajalikud nõuded. Lisaks aitab dünaamiliselt genereeritud dokumentatsioon paremini tagarakenduse API arendusi läbi viia. [63], [64] Dokumentatsioon hõlbustab keeruliste projektspetsiifiliste erisuste säilitamist ning edasiste arendajate tööd, kel puuduvad domeeniteadmised eksperimendi disaini vallas. Samuti aitab see kaasa tulevikus planeeritavatele arendustele teistes ValChromi teenustes, mis kasutavad eksperimendi disaini tagarakenduse otspunkte lisainfo saamiseks.

Tabel 7. Tagarakenduse raamistiku valimine.

Raamistik	Eelised	Puudused
Flask [65]	<ul style="list-style-type: none"> ▪ minimalistlik ▪ sobilik mikroteenuse API arenduseks ▪ suur kommuun 	<ul style="list-style-type: none"> ▪ otspunktide dokumentatsiooniks lisateegi vajadus
Django [66]	<ul style="list-style-type: none"> ▪ väga võimekas ▪ suur kommuun 	<ul style="list-style-type: none"> ▪ liiga võimekas ▪ ei sobi väikese mikroteenuse jaoks
FastAPI [63]	<ul style="list-style-type: none"> ▪ minimalistlik ▪ sobilik mikroteenuse API arenduseks ▪ otspunktide lihtne dokumenteerimine OpenAPI standardi järgi 	<ul style="list-style-type: none"> ▪ väiksem kommuun

Eksperimendi disaini jaoks vajalike arvutuste läbiviimiseks vaadeldi populaarseimaid Pythoni teeke. Tabel 8 toob võrdluse eksperimendi disaini arvutusi võimaldavatest teekidest. Valikul olid olulisteks kriteeriumiteks teegi võimekus toetada nõuetes ettenähtud ja ka tulevikus planeeritavaid arendusi ning teegi elujõulisus. Nendest kriteeriumitest lähtuvalt võeti kasutusele *pydoe2* ja *statsmodels* teegid.

Pydoe2 teek võimaldab mitmekesiseid eksperimendi disaini kavandada ning toetab sellega ka tulevikuarendusi, mitte ainult minimaalset planeeritavat teostust. *Statsmodels* teek võimaldab eksperimendi disaini toorandmete pealt väga paindlikult mudeleid ehitada ning saadud mudelite detailseid statistilisi analüüse, toetades samuti ka tulevikuarenduste võimekust.

Tabel 8. Eksperimendi disaini arvutuste jaoks sobilik teekide võrdlus.

Teek	Eelised	Puudused
Doepy [39]	<ul style="list-style-type: none"> ▪ FFD, FrFD, BBD, mitmeid teisi disaine 	<ul style="list-style-type: none"> ▪ uuendus: 2020. ▪ vähe dokumentatsiooni ▪ kasutab ise pydoe teeki
Pydoe [37]	<ul style="list-style-type: none"> ▪ FrFD, FFD, BBD, mitmeid teisi disaine ▪ hea dokumentatsioon 	<ul style="list-style-type: none"> ▪ ei ole aktiivselt arendatav, mistõttu tehti eraldi haru pydoe2 (2015)
Pydoe2 [41]	<ul style="list-style-type: none"> ▪ FrFD, FFD, BBD, mitmeid teisi disaine ▪ parandas pydoe puudused 	<ul style="list-style-type: none"> ▪ uuendus: 2020.
Dexpy [38]	<ul style="list-style-type: none"> ▪ FFD and FrFD, mitmeid teisi disaine ▪ disaini efektiivsuse hindamist 	<ul style="list-style-type: none"> ▪ uuendus: 2017. ▪ puudu BBD tugi
Doegen [40]	<ul style="list-style-type: none"> ▪ FFD ▪ disaini analüüsi võimekus 	<ul style="list-style-type: none"> ▪ ainult FFD
Statsmodel [67]	<ul style="list-style-type: none"> ▪ ANOVA võimekus ▪ võimekam ▪ paindlikud analüüsid ▪ detailed analüüsitulemused 	<ul style="list-style-type: none"> ▪ keerukam alustada
SciPy [61]	<ul style="list-style-type: none"> ▪ ANOVA võimekus ▪ lihtsam 	<ul style="list-style-type: none"> ▪ väike tulemuste detailsusaste
Scikit-learn [68]	<ul style="list-style-type: none"> ▪ lihtsam kohe alustada 	<ul style="list-style-type: none"> ▪ väike tulemuste detailsusaste

3.5 Klientrakenduse raamistik

Klientrakendus on veebirakenduse osa, mis vastutab kasutajatele andmete kuvamise ja kasutajatelt sisendi kogumise eest. Klientrakendused põhinevad baastehnoloogiatel nagu HTML, CSS ja JavaScript, kuid nende arenduseks on loodud mitmeid teeke ja raamistikke, mis kiirendavad arendusprotsesse ja hõlbustavad koodi haldamist suuremahulistes projektides. [46], [69]

Klientrakenduse raamistiku valikul lähtutakse esmalt eksperimendi disaini tulemuste kuvamise võimekusest. Vaatluse alla võeti populaarseimad JavaScripti raamistikud, milleks on *Stack Overflow* 2022. aasta küsitluse [70] põhjal VueJS, React ja Angular. React on tegelikult Meta poolt välja arendatud JavaScripti teek, kuid ühtluse huvides käsitleme seda edasi raamistikuna. Tabel 9 toob kokkuvõtte antud võrdluse tulemustest. Kuna kõikide raamistike puhul on võimalik kasutada keerulisemate interaktiivsete graafikute loomiseks sobivaid teeke, nagu D3.js [71] või Plotly.js [72], siis valiti React peamiselt väiksema rakendusega alustamise lihtsuse (autoril oli juba kogemus), hea dokumentatsiooni ja populaarsuse tõttu.

Tabel 9. Klientrakenduse raamistiku valimine.

Raamistik	Eelised	Puudused	Kogemus
VueJS [73]	<ul style="list-style-type: none"> ▪ olemasolevas tarkvaras kasutusel ▪ Typescript tugi ▪ saab juurde lisada võimekust vastavalt vajadusele 	<ul style="list-style-type: none"> ▪ pigem suunaga suurematele rakendustele 	<ul style="list-style-type: none"> ▪ keskmine
React [74]	<ul style="list-style-type: none"> ▪ väga võimekas ▪ suur kommuun ▪ üks eelistatumaid ▪ jätab palju vabadust 	<ul style="list-style-type: none"> ▪ raskem alustada, kui pole varem kogemusi 	<ul style="list-style-type: none"> ▪ keskmine
Angular [75]	<ul style="list-style-type: none"> ▪ väga võimekas ▪ suur kommuun 	<ul style="list-style-type: none"> ▪ pigem suurematele rakendustele ▪ jäigemad arhitektuurinõuded 	<ul style="list-style-type: none"> ▪ väike

3.6 Klientrakenduse toetavad teegid

Järgnevalt valiti välja sobivad klientrakenduse teegid saavutamaks rakenduse põhifunktsionaalsused. Vajalikul tasemel 2D ja 3D graafikute interaktiivsuse andis *Plotly.js* teek, millele on tugi ka React teegis eraldi komponendi *react-plotly* näol, lihtsustades sellega arendust [76]. Teistest vaadeldud teekidest *Three.js* on liiga orienteeritud 3D mudelitele mitte graafikutele [77]. *Vis.js* on fokuseeritud enam graafidele, kuigi võimaldab ka osasid vajalikke 3D graafikuid [78]. *D3.js* funktsionaalsus

asetus võrdlusele liialt võimeka teegi poolele, mis võimaldab enam kui vaja (nt animatsioonid) ning omab suuremat õpikõverat kui Plotly [71].

Andmete impordiks ja ekspordiks leiti mitmeid sobivaid teeki: *react-dropzone*, *papa-parse*, *react-csv*. Kasutusele võeti *papaparse* teek [79], mis võimaldab nii kopeeritud andmete kleepimist, failist importi kui ka tabelite andmete ekspordi. Samuti on teegil olemas automaatne erinevate failiformaatide äratundmine ja haldamine. Tabelite kuvamiseks leiti samuti mitmeid teeki (*react-table*, *react-data-grid*, *handsontable*, *react-bootstrap-table2*), mille seast võeti kasutusele teek *react-table* [80], mis jätab kasutajale oluliselt vabamad käed stiilireeglite osas, sobides antud juhul hästi kasutuselevõtuks projekti, kus stiil tuli ühtlustada olemasoleva tarkvaraga.

Üldiselt hoiti rakenduse stiil lihtsana ja ei seotud ühegi konkreetse klientrakenduse disainraamistikuga, ent kohati kasutati vastavalt vajadusele elemente *material ui* teegist. Arenduse hõlbustamiseks võeti kasutusele ka TypeScripti teek, kuna autor on enam harjunud kirjutama tüübikindlates keeltes.

3.7 Arendust abistavad tööriistad

Integreeritud arenduskeskkonnaks valiti IntelliJ IDEA, kuna autorile oli see ka tööalaselt enim tuttav töövahend ning üliõpilase litsentsiga oli ligipääs tasulisele versioonile olemas. Koodi halduskeskkonnaks valiti Git. Algul oli koodihoidlaks autori isiklik GitHubis asuv hoidla, kuid toodanguversioonis kasutuselevõtuks viidi üle olemasoleva rakenduse privaatsesse Bitbucket hoidlasse. Seal võeti kasutusele automaatne andmekonveier uute versioonide publitseerimiseks. Rakenduse koodi uuendamisel meeskonna hoidlasse käivitub automaatselt toimuv tõmmise kokku ehitus ja avaldamine meeskonna DockerHub konto all. Seetõttu oli kasutusel ka Docker Desktop haldamiseks arenduse käigus agiilselt testserverile vaheversioonide ehitamist ja avaldamist otse arendusarvutist.

API otspunktide käitumise testimiseks kasutas autor Postman rakendust, millega oli hea esmalt kontrollida tagarakenduse õiget käitumist. Klientrakenduse testimiseks kasutati peamiselt Firefox Developer Edition veebilehitsejat, kuna arenduse vahepeal tõestas too end selgemat ja paremat tagasisidet andva töövahendina võrreldes Google Chrome veebilehitsejaga.

Kuna tagarakenduses kasutatav FastAPI ega klientrakenduses kasutatav React ei olnud autori igapäevatöös kasutatavad tehnoloogiad, otsustati kasutusele võtta ka arendust abistavad tööriistad ChatGPT [81] (isiklik tasuline konto) ja GitHub Copilot [82] (tasuta testimise konto). Nii ChatGPT kui ka Copilot on tehisintellektil põhinevad tööriistad. ChatGPTd sai avalikult kasutama hakata 30. novembrist 2022. Copilot on kasutatav 29. juunist 2021. aastast. Arenduste alguse hetkeks oli ChatGPT kättesaadav olnud natukene üle kahe kuu, kuid juba oli kasutusjuhtudest näha suurt abi programmeerimiskeele süntaksi kiiremal õppimisel [83]. GitHub Copilot eelised piirdusid kiirema automaatse sõnalõpetusega, kuid see on pigem autori vähese kasutuskogemuse kui tööriista enda võimekuse taga.

ChatGPT peamine kasutus oli autorile vähem kogemust omavate tehnoloogiate õppimise kiirendamises. Tööriist oli suureks abiks süntaktiliste iseärasuste õppimisel ja koodi ümberkorraldamisel. Samuti oli kasu erinevate tehnoloogiate kohta üldise info pärimiseks, mille pealt edasi uurida juba veebi otsingumootorites. Erinevate arenduse käigus tulnud segaste veateadete puhul osutus ChatGPT väga kasulikuks pakkudes välja erinevaid võimalusi, mida uurida või ka kohe õige põhjuse.

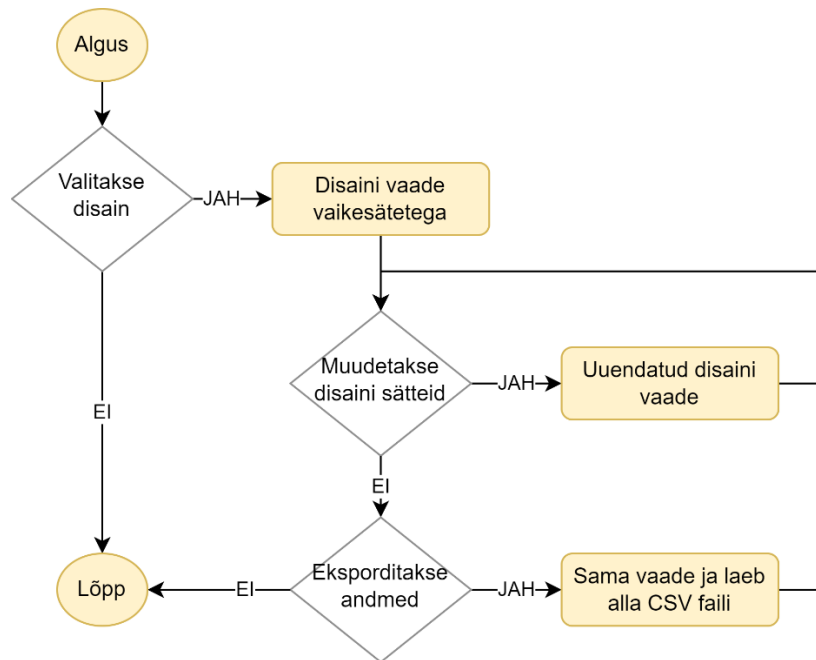
ChatGPT kasutuse puudusteks on valetave kui pärida aegunud tehnilise info kohta. Näiteks CORS poliitika iseärasusi tuleb kindlasti uurida raamistiku dokumentatsioonist, mitte uskuda, mida ChatGPT pakub. Samuti on tihti tema poolt pakutud koodifragmentides või koodi ümberkorraldustes vigu, mis takistavad koodi lihtsalt pimesi kasutusele võttu. Nende vigade ülesleidmiseks peab kasutaja omama head ülevaadet süntaktilistest erisustest. Samuti tuleb oma äriloogika kasutuseks tihti palju muudatusi teha.

3.8 Peamiste kuvade voodiagrammid

Nõuete prioriseerimise etapis (vt peatükk 3.1 "Nõuete kogumine") sai selgeks, et andmete salvestamise lahendusi (tulemuste salvestamine või raporteerimine) ei olnud antud versioonis vaja. Seetõttu ei võetud esiteks kasutusele andmebaasi, mis lihtsustas oluliselt tehnoloogiapinu. Samuti kadus vajadus kasutajakontode järele, kuna lahenduse peamiseks sisendiks on kasutaja poolt antavad toorandmed ning selle pealt arvutatud tulemuste kuvamine. Kuniks kestab veebilehitseja sessioon, hoitakse alles ka kasutaja tehtud valikud, kuid lehitseja värskendamisel kaotatakse kõik andmed. Kuna kõik

tulemuste arvutused toimuvad ühes kohas üleslaetud toorandmete pealt, on kasutajal alati võimalik kiirelt samad tulemused saada andmeid uuesti üles laadides.

Joonis 3, Joonis 4 ja Joonis 5 näitavad rakenduse kolme peamise kuva voodiagramme. Kasutajaid ei eristata, kuna puuduvad erinevad kasutustasemed. Joonis 3 kujutab disaini vaate erinevaid võimalikke tegevusi. Disaini vaate peamine eesmärk on võimaldada sobiva disaini valimine, selle parameetrite täpsustamine ning disaini eksportimine.

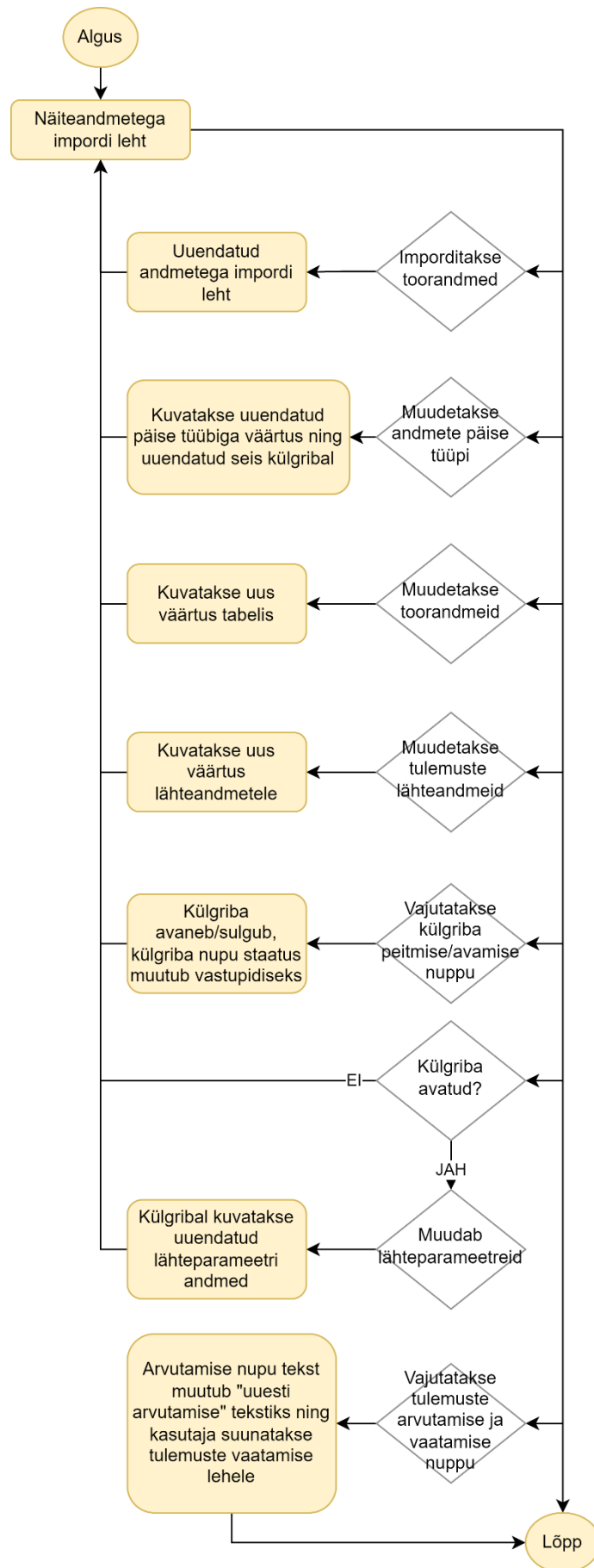


Joonis 3. Disainide kuva kasutaja tegevuste voodiagramm.

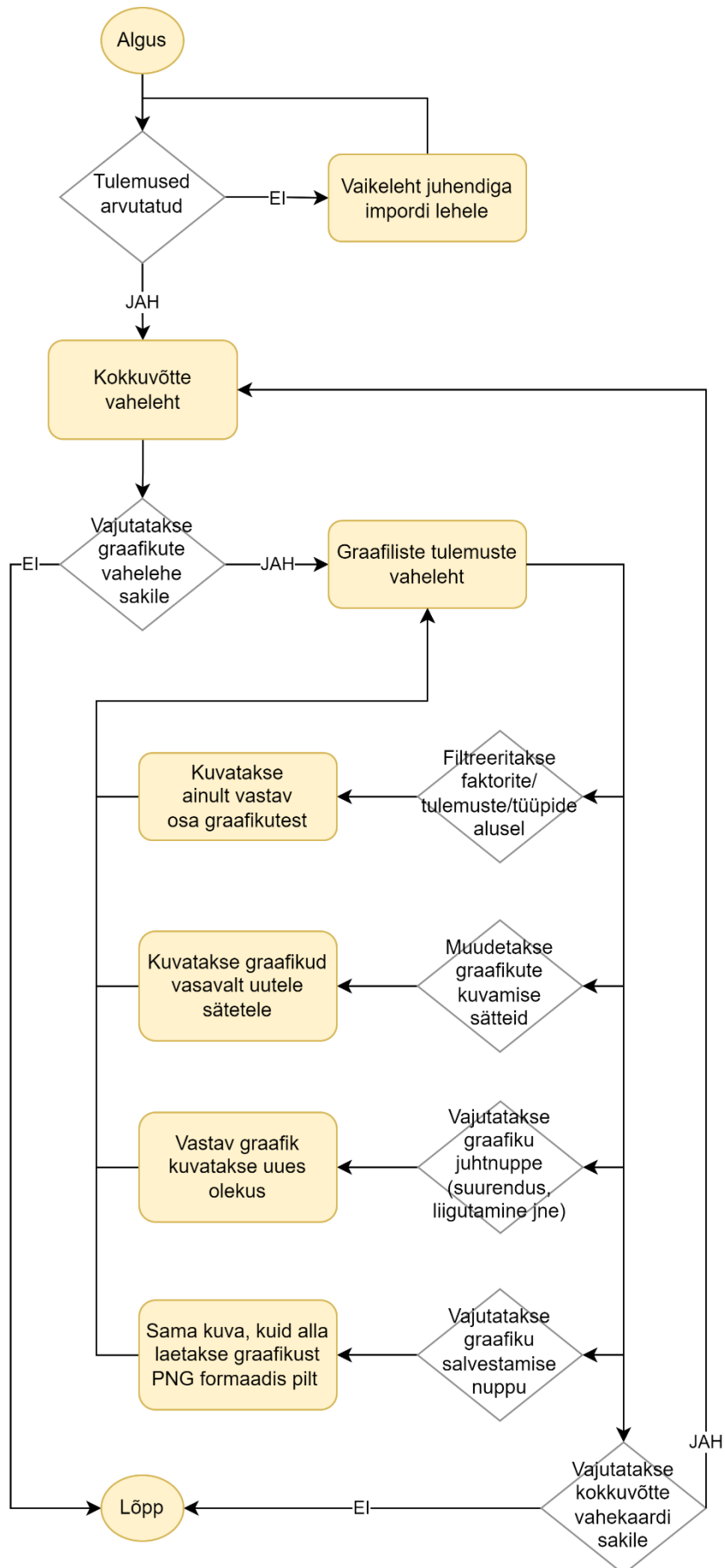
Joonis 4 kujutab impordi vaate võimalusi, mis on seotud laboris teostatud eksperimentidest saadud toorandmete impordiga. Imporditud andmete puhul on võimalik veel andmeid parandada ning tuleb täpsustada, millised andmed on faktorite omad ja millised mõõtetulemuste omad. Kui soovitud parandused on tehtud, siis on võimalik ka muuta veel arvutusteks olulisi parameetreid ning seejärel minna tulemuste kuvale.

Joonis 5 annab ülevaate tulemuste vaate võimalustest. Peamised valikud seisnevad kahe eri vaate vaatamises (kirjaliku kokkuvõtte ja graafikute kuvarid) ning graafikute vaatel erinevatel alustel filtreerimistes. Kasutaja saab tulemusi vajadusel alla laadida.

Lisaks toodu voogudele on kõikidel lehtedel ka tagasiside andmise voog, avades vastava nupu abil tagasiside vormi. Tagasiside saadetakse vastavale tagarakenduse otspunktile, misjärel saadetakse meeskonna mailiaadressile vastavasisuline kiri.



Joonis 4. Impordi kuva kasutaja tegevuste voodiagramm.



Joonis 5. Tulemuste kuva kasutaja tegevuste voodiagramm.

4 Lahenduse loomine

Lahenduse loomist teostati agiilselt, alustades algselt kriitilistest funktsionaalsustest, et saada aimu, mil määral on nõuded täidetavad. Paralleelselt arendati nii tagarakendust kui ka klientrakendust, viies kummalgi pool sisse uue funktsionaalsuse muudatused. Seejärel arendati olulisemad funktsionaalsused, samal ajal iganädalastel koosolekutel seisu raporteerides ning lahtiseid küsimusi täpsustades.

Kui olulisemad funktsionaalsused said omavahel ühendatud ühtseks kasutusvooks, pandi rakendus üles veebikursuse meeskonna testserverile, lubades ligipääsu läbi olemasoleva rakenduse. Nõnda said veebikursuse korraldajad testida otse keskkonnas, mis oli juba tuttav ja lõpliku keskkonna koopia. Seejärel kohtuti eraldi kuvade prototüüpimiseks, mis teostati tahvlil arutelu käigus. Tahvliprototüübid viidi sisse juba otse rakendusse ning iganädalaselt avaldati uus versioon testserveris korraldajatele testimiseks.

4.1 Tagarakenduse arendus

Tagarakenduse arendust alustati esmalt lihtsustatud versiooniga, milles oli üks üldine disain (FD), lihtne navigeerimine erinevate lehtede vahel ja tulemuste mudeli koostamine staatiliste andmete pealt. Eksperimendi disaini andmete pealt arvatav mudel saadakse vähimruutude meetodil [84] ning on rakenduse sõlmpunktiks, sidudes ära disainist saadud toorandmed ja kuvatavad tulemused, mistõttu selle sõlmpunkti katsetused olid esmatähtsad.

Tagarakenduses kasutati testide kirjutamist *pytest* teegi abil, et alguses kiirelt muutuvasse koodi ei satuks liialt ulatuslikke vigu sisse. Testide kirjutamine oli kasulik eriti esmaste meetodite ja statistiliste mudelite väljatöötamisel. Need aitasid üheaegselt läbi mõelda nii andmete struktuuri kui ka hoiduda rakenduse liialt katki tegemise eest, kui katsetati erinevaid statistiliste arvutuste meetodeid.

Järgnevalt prooviti läbi variandid tulemuste info edastamiseks, katsetades nii faktoritekeskset andmemudelit kui tulemustele orienteeritud mudelit. Paralleelse

klientrakenduse arendusega selgus, et esialgu kasutusele võetud faktorikeskne andmemudel ei ole sobiv, kuna nõuab klientrakenduse pool palju ümberorienteerimist tulemuste kuvamiseks vajalike andmestruktuuride saavutamiseks. Klientrakenduses kuvatavate tulemuste kontekst on fokuseeritud just erinevate mõõtetulemuste ja neid mõõtetulemusi mõjutavate faktoritele suunal, mitte ühe faktori mõjule erinevatele mõõtetulemustele.

Seejärel keskenduti mudelist väljasaadavale infole, mis tõi välja puuduse esialgu valitud mudeli koostamise teegis *scikit-learn* [68], milles polnud võimalik põhjalikum mudeli analüüsi osa. *Scikit-learn* teek sai esialgu valitud mudeli koostamiseks, kuna autoril oli sellega varasemalt kogemus ja olemas koodinäited, mille pealt esialgne lihtne versioon tekitada ning testida. Seejärel oli selgem, mis on vajalik ja võimalik. Teostati põhjalikum analüüs ning valiti välja *statsmodels* teek, mille kasutuselevõtmine oli oluliselt keerulisem, kuid mudeli kohta saadav statistiline info oluliselt detailsem.

Olles lahendanud kriitilisemad funktsionaalsed lahendused, loodi testserverisse evitamise tugi konteinerite ehitamiseks vajaliku tõmmise ja keskkonna parameetrite erisuste osas. Selleks võeti kasutusele eraldi keskkonnamuutujate fail, millega sai kontrollida keskkonna erisusi arendus-, test- ja toodanguserveritel. Agiilselt lahendati klientrakenduse arendusest ja kursuse korraldajate tagasisidest saadud kuvade jaoks vajalikke muudatusi.

Üks nõudeid, mis lisandus hiljem arenduste käigus, oli kasutajate tagasiside saatmine meeskonna ühisele e-mailile. Kasutaja tagasiside saadetakse kasutades Pythoni *smtplib* teeki, mis lubab SMTP protokolliga kasutades saata elektroonilisi kirju, kui ette anda vastavad konfiguratsioonid (emaili aadress, sisselogimise info, sisu vormindus jne). Konfiguratsiooni tundlikumad osad hoiti koodist eraldi keskkonnamuutujate failis.

Viimaks täpsustati OpenAPI standardi kohaselt esimese versiooni otspunktide dokumentatsioon. Antud tegevuseks on FastAPI raamistikus sisseehitatud tugi, mis tegi otspunktide kirjeldamise lihtsamaks. Tagarakenduse otspunktide kirjelduseks tekitab FastAPI eraldi otspunkti `/docs`, mille pealt saab infot kõikide kättesaadavate otspunktide, saadetava ja saadava info näidete ja kirjelduste kohta.

4.2 RESTful API tagarakenduse liides

Tagarakenduse arendusel lähtuti RESTful API põhimõtetest [85], [86]. API on veebiteenuse liides, mis reguleerib erinevate komponentide omavahelist suhtlemist, määrates ära vajalikud elemendid ja reeglid [43]. REST (*Representational State Transfer*) on arhitektuuriliste piirangute komplekt, mis esmalt pakuti välja Roy Fieldingu poolt [87], [88], tagamaks paremat veebirakenduste arendust ja haldamist. REST piirangud ei määra ära ressursside edastamise vormi ega protokollit, see valik jääb arendaja teha. RESTful API on liides, mis rakendab REST põhimõtteid. [89]

Tabel 10 näitab tagarakenduses kättesaadavaid otspunkte. API dokumentatsioon on kättesaadav vastavas otspunktis /docs. Otspunktide koostamisel lähtuti parimatest praktikatest hoidmaks otspunktide aadressid selged [90]. Otspunktide parameetrite puhul on näidatud ka vaikeväärtused, juhuks kui API kasutaja vastavat väärtust ei tagasta.

Tabel 10. Versioon 1 API otspunktid.

Ressurss	Päringu tüüp	Päringu kirjeldus
/designs	GET	Otspunkt võimalike disainide pärimiseks
/designs/pbd?factors={11}	GET	PBD disaini päring
/designs/frfd?factors={puudub}&confounded{0}&generator={puudub}	GET	FrFD disaini päring
/designs/fd?factors={3}&levels={3}	GET	FD disaini päring
/designs/bbd?factors={3}¢ers={3}	GET	BBD disaini päring
/results	POST	Tulemuste päring vastavalt üleslaetud toorandmetele
/feedback	POST	Kasutaja tagasiside saatmine
/docs	GET	API dünaamiline dokumentatsioon

API otspunktide sisendandmete valideerimine toimub kahetiselt. Lisaks ärioloogikale vastavale sisendandmete kontrollile, teostab ka FastAPI raamistik automaatselt sisendandmete valideerimist. Vastavad standardsed HTTP veakoodid, koos lubatud väärtuste ja näidetega on API dokumentatsioonis kirjeldatud.

Joonis 6 näitab üht tagarakenduse otspunkti. Sealt päritakse Plackett-Burman disain vastavalt kasutaja määratud faktorite arvule. FastAPI raamistik võimaldab lihtsalt defineerida limiidid sisendparameetritele, näites defineeritud suurem või võrdne (ge) ning väiksem kui võrdne (le) väärtused. Samuti haldab ühine *router* objekt teiste otspunktidega ühist käitumist (ühine otspunkti segment /v1/designs, sama veakäitumise kirjeldus jne).

```

@router.get("/pbd",
            tags=["screening", "plackett-burman design"])
@limiter.limit("1000/minute")
async def get_pbd_design(
    request: Request,
    factors: int = Query(
        11,
        description=f"Number of factors "
                    f"({PBD_MIN_FACTORS}-{PBD_MAX_FACTORS})",
        ge=PBD_MIN_FACTORS,
        le=PBD_MAX_FACTORS)
) -> Design:
    try:
        return {
            "title": f"Plackett-Burman design for {factors} factors.",
            "design": jsonable_encoder(pbdesign(factors).tolist()),
            "factors": f"{factors}"
        }
    except ValueError as ve:
        logger.warning(
            f"ValueError in generating pbd design "
            f"with factors '{factors}': {ve}. "
            f"Possibly encoder error.")
        raise HTTPException(
            status_code=500,
            detail="Design generation error. "
                  "Try again later. "
                  "If problem persists contact support.")
    except Exception as e:
        logger.warning(
            f"Unexpected exception in generating pbd design "
            f"with factors '{factors}': "
            f"{e}")
        raise HTTPException(
            status_code=500,
            detail="Unexpected server error. "
                  "Try again later. "
                  "If problem persists contact support.")

```

Joonis 6. Näide otspunktide defineerimisest FastAPI raamistikus.

4.3 Klientrakenduse arendus

Klientrakendus arendati tagarakendusega paralleelselt, saamaks kohest tagasisidet, millised tagarakenduse andmemudelid ja otspunktid on vajalikud. Arendus Reactis

toimus vastavalt komponentidel põhinevale arhitektuurile (CBA – *Component-Based Architecture*), kus rakenduse kasutajaliides jagatakse väiksemateks iseseisvateks komponentideks. Komponentid sisaldavad kõiki muudest komponentides eraldiseisvalt uuendamiseks vajalikke osi (stiilireeglitest äri loogikani) [91], [92]. Komponente kirjutatakse JSX (JavaScript XML) süntaksis, mis on HTML-laadne kood, mis teiseneb lõpuks JavaScripti koodiks [93].

Klientrakendus loodi kasutades *Create React App* tööriista, mis aitab lihtsalt ja kiirelt uue Reacti projekti püsti panna. Eelnev konfigureerimine tehakse tööriista poolt ära, kasutades teisi tööriistu (*webpack*, *babel*, jm) ning samuti on võimalik lihtsalt Typescript keele tugi kohe alguses kaasata. *Create React App* kasutuseks oli vajalik arendusmasina peale installeerida arendusserverit pakkuv ja paketi haldust ning ehitusprotsessi lihtsustav tööriist *Node*. [94]

Klientrakendus arendati üheleherakendusena (SPA, *single-page application*), mida Reacti komponentideks jaotamise loogika hästi toetab. SPA puhul uuendatakse ainult komponente, mis muutuvad kasutaja tegevuste mõjul, lubades nõnda paremat jõudlust ja pääseda kogu lehe uuesti laadimisest. [95] Navigeerimine klientrakenduses lahendati kasutades *react-router* teeki [96].

Esmalt keskenduti teekidele, mis võimaldasid soovitud tasemel graafikuid kuvada, kasutades esialgseid tagarakenduse poolt pakutavaid staatilisi andmeid. Algul arendati rakendus ühes failis ning vastavalt vajadusele hakati sellest tekitama eraldiseisvaid komponente, kuna kohe ei olnud selge, milliseid komponente on vaja. Kasutades „otsusta võimalikult hilja“ (*defer commitment*) lähenemist [97], lükati otsused komponentide arhitektuuris edasi, võimaldades kiiremini muutustele kohaneda vastavalt arenduse käigus kogunenud lisainfole. Kui arenduse käigus selgusid, millised osad on mitmes kohas vajalikud, eraldati vastav osa eraldiseisvaks komponendiks. Näiteks on toodud eksperimendi disainide valikulehte haldav komponent (Joonis 7). Algselt oli komponendis kogu ühe disaini kuvamise loogikat, kuid hilisemate disainide lisamisel muutus selgelt liiga mahukaks ning korduvaks, mistõttu leiti ühisosa ning koostati uus komponent (joonisel rõhutatult *DesignOption*), mille abil oli lihtne sama korduvat koodi lisada. Joonisel on jäetud kuvamise ja selguse eesmärgil pikemad tekstid ja sama loogikaga osad välja ning asendatud kolme punktiga.

```

return (
  <div className="designs-container">
    <div className="designs-section">
      <h2 className="section-title">Screening Designs</h2>
      <div className="designs-group">
        <DesignOption
          className={ "pbd" }
          imageSrc={ pbdImage }
          altText={ "Plackett-Burman Designs" }
          linkTo={ "/designs/plackett-burman" }
          title={ "Plackett-Burman Designs" }
          description={ "Plackett-Burman designs ..." }
          furtherReadingUrl="https://www.itl.nist.gov/..."
        />
        <DesignOption
          className={ "frfd" }
          imageSrc={ pbdImage }
          ...
        />
      </div>
    </div>
    <div className="designs-section">
      <h2 className="section-title">Optimization Designs</h2>
      <div className="designs-group">
        <DesignOption
          className="fd"
          imageSrc={ fdImage }
          ...
        />
        <DesignOption
          className="bbd"
          imageSrc={ bbdImage }
          ...
        />
      </div>
    </div>
  </div>
);

```

Joonis 7. Näide klientrakenduses komponentideks jagamisest.

Pärast esialgsete olulisemate graafikute kuvamise võimekuste läbitestimist keskenduti andmete impordile ja selle sidumisele graafikute kuvamisega. Järgnevalt lisati täpsemate statistiliste andmete kuvamise loogikad ja täiendavad disainid. Ühtlasi iga järgneva muudatuse käigus eraldati ühtsest koodist aina selgemini eristuvad iseseisvamad komponendid.

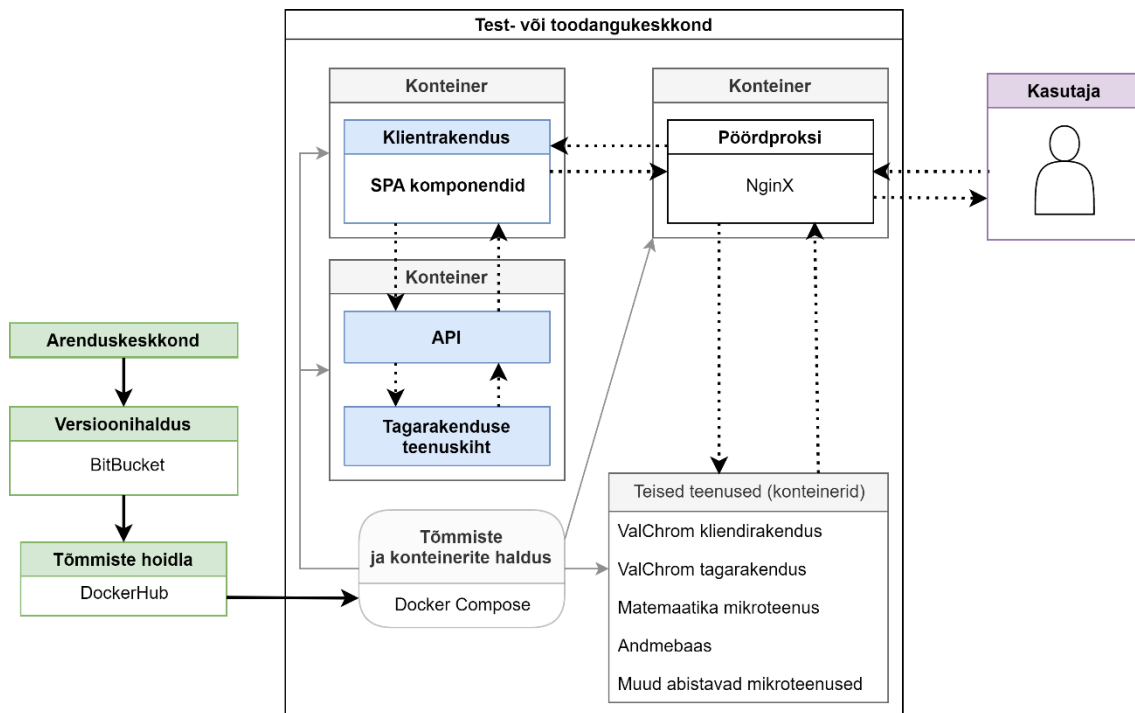
Kuvade prototüübid koostati koos kaasjuhendajaga eraldi koosolekul, kasutades testserveris olevat hetkeseisu. Vastavalt arutelule koostati jooksvalt tahvliprototüübid. Antud prototüüpide pealt viidi muudatused sisse koheselt testserveris olevasse rakendusse ning arutati jooksvalt vastavalt vajadusele uusi muudatusi. Kuna kuvasid oli vähe ja pigem lihtsakoelised, ei hakatud eraldi aega kulutama lisatarkvara kasutusele, et teha täpsemaid prototüüpe.

Klientrakenduse puhul oli üheks olulisemaks väljakutseks vähendada kliendi poolt graafikute filtreerimisel ooteaega. Kuna graafikuid on palju ja nende arv on otseselt seotud kasutaja poolt üles laetavate andmemahutudega, siis viidi sisse filtreerimise võimekus, et kasutaja saab parasjagu vaadelda ainult konkreetset huvipakkuvaid graafikuid. Algselt tuli iga graafiku kuvamise sisse- ja väljalülitamise järel oodata komponenti järgi jäänud graafikute uuesti kuvamist mitu sekundit, mis ei ole mugava kasutuskogemuse põhimõtete kohane. Kasutaja tajub sujuva ja koheselt reageeriva veebilehena muudatusi, mis jäävad alla 100 ms [98]–[100]. Erinevate graafikute vahel liikumine ja võrdlemine peab olema võimalikult kiire ja mugav, et mitte hoida tagasi kasutaja mõttelõnga tulemuste tõlgendamisel. Samas tulemuste esmane mõnevõrra aeglasem laadimine on kasutuskogemuse poolt vastuvõetavam (möödetav sekundites, kus neli kuni seitse sekundit on kehv ja üle seitsme sekundi juba halb kasutajakogemus) [101]. Seetõttu otsustati kõik graafikud komponendi kuvamise esimesel hetkel ära laadida ning edasine filtreerimine lahendada juba graafikutele staatilise CSS reeglistikuga lahendada.

4.4 Pidev integratsioon ja evitamine

Arendatud mikroteenus tuli juba olemasolevasse lahendusse sobitada. Pidev integratsioon põhineb Bitbucketi andmekonveieril, mille põhiharusse koodi mestides käivitub automaatne konteineri tõmmise kokku ehitamine. Tõmmis on tarkvara koodi põhjal kokku ehitatud tarkvarapakett, mis sisaldab kõike vajalikku sees oleva tarkvara käivitamiseks (peale koodi näiteks ka vajalikud teegid, käivituskeskkonna jaoks vajalikud vahendid jm). Seejärel laaditi tõmmis üles meeskonna privaatsele DockerHub kontole, kust sai seda vastavalt vajadusele alla tõmmata. Test- ja toodanguserveril käis uuendamine käsitsi *Docker Compose* tarkvara kasutades [102], tõmmates esmalt uue tõmmise versiooni ja seejärel käivitades evitamist kirjeldava YAML faili uuendused. Selle tulemusel luuakse uut tõmmist kasutades käivitatavad instantsid (konteinerid),

vahetades välja vanad versioonid. [103]. Eksperimendi disaini teenuse sobitumist vastavasse arhitektuuri näitab Joonis 8.



Joonis 8. Eksperimendi disaini teenuse paiknemine muude komponentide kontekstis.

Tagamaks automaatse toimimise olenemata keskkonnast võeti klientrakenduses kasutusele Reacti sisseehitatud keskkonnamuutujate haldus, mis lubas vastavalt defineerida *.env.development* ja *.env.production* failides eri keskkonnades jooksutamiseks vajalikud muutujad [104]. Eri keskkonnamuutujates kajastati erinevused tagarakenduse päringu aadressis ning test- ja toodangukeskkonnas kasutusel oleva pöördproksi (*reverse proxy*) taga toimimiseks vajalike muutujatega. Pöördproksi vastutab tarkvara veebiliikluse suunamise eest vastavale rakenduse konteinerile [105].

Eespool olev kirjeldus kehtib nii test- kui ka toodanguserverisse integreerimise ja evitamise protsessile. Peamine erinevus arenduskeskkonnaga oli automaatne konteinerite tõmmiste kokku ehitamine, mis toimus arendaja arvutis käsitsi vastavalt vajadusele saamaks uued versioonid testserverisse.

4.5 Turvalisus

Klientrakenduse raamistikud aitavad küll turvalisust tõsta, kuid ei tee automaatselt rakendust turvaliseks [106]. Seetõttu rakenduse turvalisuse tõstmiseks võeti vaatluse alla OWASP Top 10 ohtude nimekiri [107] ning analüüsiti vastavaid ohtusid antud rakenduse kontekstis. Sisse viidi järgnevad lahendused:

- API ligipääsu võti;
- sisendi puhastamine;
- päringulimiidid API otspunktidele;
- CORS (*Cross-Origin Resource Sharing*) poliitika.

API ligipääsu võti lubab päringuid API pihta teostada ainult võtit omavatel rakendustel. Sellega piiratakse kõrvaliste kolmandate osapoolte päringuid API otspunktide pihta, võimaldades ka eristada erinevaid rakendusi erinevate võtmetega ning seeläbi omada ülevaadet API kasutuse üle. [108]

XSS (*Cross-Site Scripting*) rünnak seisneb veebirakenduse sisendivälja kaudu kahjuliku koodi sisestamises, mis võib laadida ohvri veebilehitsejasse sisu, mida seal algselt polnud. XSS rünnak võib põhjustada andmete vargust, kontode ülevõtmist, kasutaja suunamist võltslehtedele ja muud kahjulikku tegevust. [107] Sisendi puhastamine ennetamiseks XSS rünnakuid teostati klientrakenduse poolel *dompurify* teegi abil [109], mis eemaldab kahjulikud HTML elemendid sisendi sõnedest. Tagarakenduse poolel kasutati samal eesmärgil teeki *bleach*. Samuti viidi sisse valideerimised kõigile API parameetritele, kasutades FastAPI sisseehitatud võimekust.

Jõurünnakul (*brute-force*) õngitsusrünnaku kaudu võidakse üritada ära arvata API ligipääsuks kasutatavaid võtmeid. Samas ummistusrünne (DDoS, *Distributed Denial of Service attack*) takistaks teenuse normaalset toimimist serveri ülekoormamisega suure hulga päringutega. Mõlemat ohtu on võimalik vähendada viies sisse päringute limiteerimise võimekus kõikidele tagarakenduse otspunktidele, mille implementeerimiseks kasutati *slowapi* teeki. [107], [110]

CORS poliitika eesmärgiks on turvalisuse tagamine, määrates ära millistelt domeenidelt on lubatud päringuid teha vältimaks volitamata ressursi juurdepääsu. CORS poliitika aitab vähendada CSRF (*Cross-Site Request Forgery*) rünnakuid, mille käigus ründaja

petab ohvri veebilehitsejat ohvri nimel volitamata tegevuste sooritamiseks (privaatsete andmete muutmine, lekkimine jms). Rakenduses sätestati CORS poliitikas lubatud domeeniks ainult vastav ValChrom domeen, keelates muud ligipääsud. [107]

Kokkuvõtvalt lähtuti turvalisuse tagamisel levinud ohuvektoritest, mis kirjeldatud OWASP lehel Top 10 nimistus. Kõikide võimalike ohuvektorite neutraliseerimine ei mahtunud antud töö skooopi ega pole antud töö autori pädevuses. Oluliselt suuremal määral turvalisuse tõstmiseks on vaja kaasata turvatestimise spetsialist ning turvalisuse tõstmine on valdkond, millega tuleb pidevalt edasi tegeleda.

5 Loodud tarkvara

Antud peatükis kirjeldatakse valminud rakendust. Rakendus koosneb kolmest peamisest osast: disainide, importimise ning tulemuste vaated. Disaini valiku lehel (Joonis 9) kuvatakse saadaolevaid meetodeid kahes osas: sõelumine ja optimeerimine. Kummagi alt on võimalik valida kahe disaini vahel. Igal disainil on lühike kirjeldus ning lingiga viide välisele avalikult kättesaadavale allikale, kus on põhjalikud sissejuhatused vastavasse eksperimendi disaini meetodisse. Lisaks on Joonisel 9 all paremal nurgas nähtav ka tagasiside saatmise kuva, mis on avatav kõikidel vaadetel (vaikimisi suletud).

The screenshot displays the 'Design of Experiments' software interface. At the top, there is a header with the 'VolChrom' logo and the text 'Design of Experiments'. On the right side of the header, there are dropdown menus for 'Sections' and 'Menu'. The main content area is divided into two columns: 'Screening Designs' and 'Optimization Designs'. Under 'Screening Designs', there are two options: 'Plackett-Burman Designs' and 'Fractional-Factorial Designs'. Under 'Optimization Designs', there are two options: 'Factorial Designs' and 'Box-Behnken Designs'. Each option includes a small 3D plot icon and a brief description. At the bottom right of the main content area, there is a 'Feedback' section with a text input field and a 'Submit' button. The footer contains the copyright information '© Tartu Ülikool/University of Tartu 2023' and a license link 'License (CC BY NC SA 3.0)'. A blue envelope icon is also present in the bottom right corner.

Joonis 9. Disaini valimise vaade.

Valides vastava disaini meetodi, suunatakse kasutaja disaini parameetrite muutmise ja disaini ekspordi lehele (Joonis 10). Muutes disaini parameetreid, saadakse uus tabel lehe paremal küljel, mis kirjeldab antud disaini ning mille saab ekspordida CSV faili kujul, et seda edasiste laboriekperimentide planeerimisel kasutada.

Factors: 5
Confounded factors: 1

OR define the factors and confounding effects yourself

Enter generator... >

Export to CSV

Design generator: a b c d ab

#	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
1	-1	-1	-1	-1	1
2	1	-1	-1	-1	-1
3	-1	1	-1	-1	-1
4	1	1	-1	-1	1
5	-1	-1	1	-1	1
6	1	-1	1	-1	-1
7	-1	1	1	-1	-1
8	1	1	1	-1	1
9	-1	-1	-1	1	1
10	1	-1	-1	1	-1
11	-1	1	-1	1	-1
12	1	1	-1	1	1
13	-1	-1	1	1	1
14	1	-1	1	1	-1
15	-1	1	1	1	-1
16	1	1	1	1	1

© Tartu Ülikool/University of Tartu 2023
License: CC BY-NC-SA 3.0

Joonis 10. Disaini parameetrite muutmise ja ekspordi vaade.

Kui tarkvaraväliselt on laboriekspriimendid sooritatud, järgneb toorandmete üleslaadimise vaade (Joonis 11). Seal on juba esialgsed näiteandmed olemas, kuid kasutaja saab need lihtsalt asendada uute andmetega. Seejärel saab kasutaja märkida, millised on faktorite tulbad ja millised on mõõtetulemuste tulbad, vastava päise peale klõpsates. Päis muudab värvi ning vasemal külgribal uueneb faktorite ja mõõtetulemuste nimekiri. Lisaks saab kasutaja kuvatavas tabelis ka otse andmeid muuda. Klõpsates tulemuste arvutamise nupule, päritakse tagarakendusest toorandmetele vastavad tulemused ning suunatakse kasutaja tulemuste lehele.

Paste here

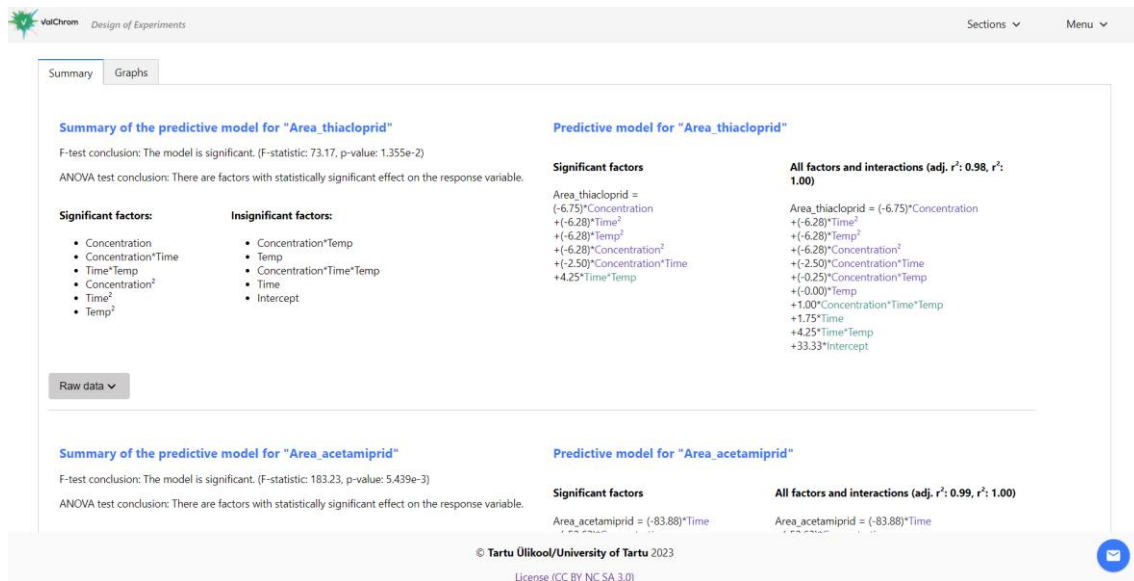
Calculate results and view

#	Factor: Concentration	Factor: Time	Factor: Temp	Response: Area_thiacloprid	Response: Area_acetamiprid
1	1	1	1	12	145
2	1	1	-1	2	221
3	1	-1	1	3	300
4	1	-1	-1	14	218
5	-1	1	1	29	234
6	-1	1	-1	22	159
7	-1	-1	1	14	400
8	-1	-1	-1	20	512
9	0	0	0	33	65
10	0	0	0	35	78
11	0	0	0	32	88

© Tartu Ülikool/University of Tartu 2023
License: CC BY-NC-SA 3.0

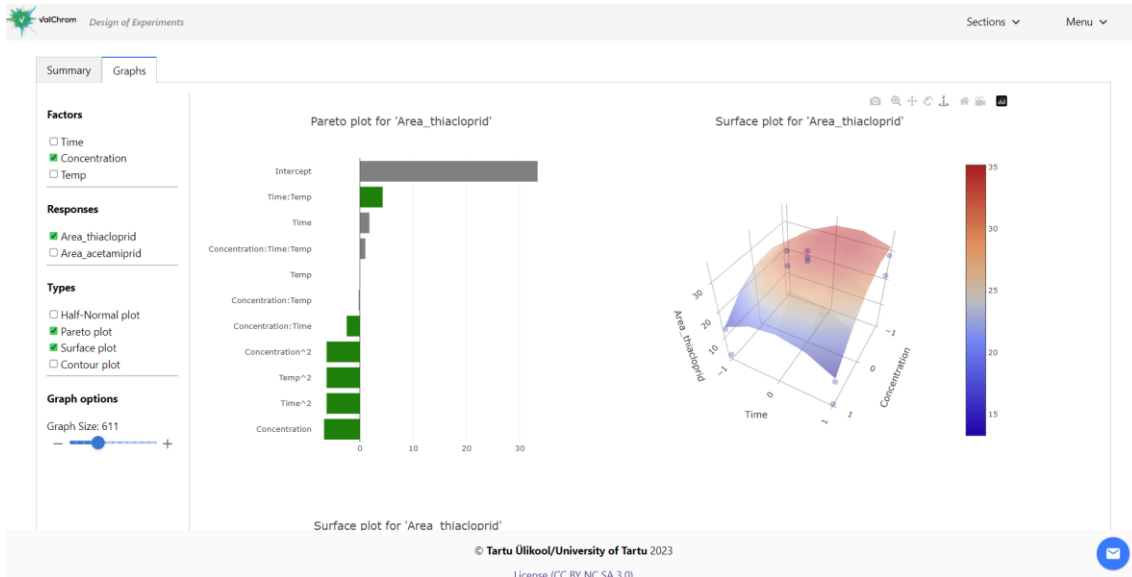
Joonis 11. Toorandmete impordi vaade.

Tulemuste kuva on jagatud kaheks vaheleheks: kirjalik kokkuvõte ning graafikud. Kirjalikul kokkuvõttelehel (Joonis 12) on toorandmete pealt loodud mudeli andmed. Välja on toodud statistiliste testide (F-test ja ANOVA) tulemused, mis määravad, kas mudel sisaldab statistiliselt olulisi efekte. Kasutaja saab kiire ülevaate sellest, millised on olulised ning mitteolulised faktorid. Avades toorandmete menüü saab kasutaja tulemusi tabelis vaadelda ja CSV faili kujul alla laadida.



Joonis 12. Disaini tulemuste kirjaliku kokkuvõtte vaade.

Graafikute kuval (Joonis 13) on toodud mudeli põhjal koostatud eri liiki graafikud, mis hõlbustavad tulemuste analüüsi. Graafikute leht võimaldab vasakpoolisel külgribal filtreerida vastavalt huvipakkuvate faktorite, mõõtetulemuste ja graafiku tüübi alusel. Samuti saab muuta graafikute suurust.



Joonis 13. Disaini tulemuste graafikute kuva.

Joonis 14 ja Joonis 15 on näited tagarakenduse dokumentatsiooni vaadetest. Dokumentatsioonis on kajastatud kõik tagarakenduse otspunktid, nende vastuvõetavad parameetrid ning parameetrite vahemikud. Dokumentatsioon võimaldab ka samal lehel erinevaid otspunkte katsetada ning saadavaid tulemusi analüüsida.

fractional factorial design

GET /v1/designs/frfd Get Frfd Design

Try it out

Name	Description
factors integer (query) maximum: 10 minimum: 2	Define either factors or generator string. Number of factors, allowed values: 2-10. <input type="text" value="factors"/>
confounded integer (query) maximum: 5 minimum: 0	Number of confounded factors, allowed values: factors - confounded > confounded. Default value : 0 <input type="text" value="0"/>
generator string (query) maxLength: 100 minLength: 3	Define either factors or generator string. Generator string, allowed symbols: a-z, space, underscore. <input type="text" value="generator"/>
x-api-key string (header)	<input type="text" value="x-api-key"/>

Joonis 14. Tagarakenduse dokumentatsiooni ühe otspunkti parameetrite näide.

Dokumentatsiooni päringuvastuste osas on kirjeldatud päringu vastused ja võimalikud veakoodid. Mõlemal juhul on näidatud vastuste struktuur ning näited, mis aitavad edasisi arendusi lihtsamini teostada. Joonis 15 näitab eduka vastuse näite dokumentatsiooni FrFD disain päringu jaoks. Samuti on näidatud mõned veakoodid ja veateate struktuur.

Responses		
Code	Description	Links
200	Successful Response	No links
	Media type <input type="text" value="application/json"/> Controls Accept header.	
	Example Value Schema <pre> { "title": "Fractional-Factorial design for 3 factors with generator 'a b ab'", "design": [[1, 1, 1, 1], [-1, 1, 1, 1], [1, -1, -1, 1], [-1, -1, -1, 1]], "factors": "3", "generator": "a b ab" } </pre>	
400	Range errors in parameters	No links
	Media type <input type="text" value="application/json"/> Example Value <pre> { "detail": [{ "Ioc": ["query", "factors"], "msg": "Wrong design parameters, acceptable values: factors 2-10", "type": "value_error" }] } </pre>	
404	Not Found, check the URL for legal routes	No links

Joonis 15. Tagarakenduse dokumentatsiooni ühe otspunkti võimalike päringuvastuste vaade.

6 Tulemused ja võimalused edasiarendusteks

Töö tulemusel valmis veebiteenus, mis vastas kõikidele prioriseerimisel püstitatud nõuetele. Rakendus võimaldab valida kahe sõelumise ja kahe optimeerimise disaini vahel, importida eksperimentidest saadud toorandmed ning teostab ja kuvab toorandmete pealt arvutatud tulemused. Bakalaureusetöö käigus loodi rakenduse esimene versioon, mis läheb kasutusele järgnevas veebikursuse „LC-MS meetodite valideerimine“ sessioonil. Seejärel jätkub ka rakenduse edasiarendus vastavalt kursusel osalejate tagasisidele. Arendatud tarkvara on kättesaadav aadressil <https://valchrom.ut.ee/doe> ning mõeldud tasuta kasutuseks nii veebikursuse osalejatele kui ka kõigile teistele huvilistele.

Arenduse käigus õnnestus sisse viia enam kui kaks esialgu plaanitud disaini. Peamiste voogude arendamise järgselt selgus, et lihtsalt sai juurde lisada ühe lisadisaini sõelumise disainide alla ja teise optimeerimise disainide alla. Nii tagarakenduse kui ka klientarakenduse arendatud struktuurid olid hästi kohendatavad uutele disainidele. Lisadisainid võimaldavad oluliselt mitmekülgsemat lähenemist õpetamisel ning ka võrdlust erinevate disainide vahel, mistõttu antud muudatus otsustati algsetele nõudmistele juurde lisada.

Rakenduse võimalikke edasiarendusi on mitmeid. Osad neist tuvastati funktsionaalsete nõuete prioriseerimise etapis (vt peatükk 3.1). Prioriteediks on võtta rakendus kasutusele järgmises veebikursuse toimumise tsüklis, kuid juba enne seda on plaan eelnevaid kursusel osalenuid sellise teenuse lisandumisest teavitada ning koguda tagasisidet. Selle põhjal arendatakse praegust versiooni edasi, et saada esimeseks veebikursuse sessiooniks parem versioon. Potentsiaalsed arendused on toodud järgnevalt:

- Täiendavad eksperimentide disaini meetodikad.
- Täiendavad statistilised analüüsid.
- Graafilise materjali kuvamise uued viisid.
- Kasutaja toorandmete valideerimine hindamaks vastavust teostatud disainiga.
- Kodeeritud faktorite info importimine kuvamaks graafikutel detailsemat infot.

- Kasutajate tagasiside põhjal kaaluda järgnevate vajadust: raporteerimised, kasutajakontod ja tulemuste salvestamise võimekus.

Lisaarendused on plaanis eksperimendi disaini mikroteenuse tihedam ühendamine põhirakenduse voogudega. Järgnevad on juba plaani võetud tulevikuarendused:

- ValChromi klientrakendus saab pärida eksperimendi disaini tagarakenduselt saadaolevad robustsuse disaini meetodid.
- ValChromi klientrakendus saab eksperimendi disaini tagarakenduselt pärida olulisemate tulemuste raporti ning siduda see põhivalideerimise raporteerimise funktsionaalsusega, saamaks automaatselt tekitatava tervikliku valideerimisraporti.

7 Kokkuvõte

Bakalaureusetöö eesmärk oli arendada mikroteenus TÜ Analüütilise keemia õppetooli rahvusvahelisele veebikursusele „LC–MS meetodite valideerimine“. Esmane versioon võimaldab enamate meetodite kasutust kui algselt plaanis oli. Samuti on täidetud ka kõik prioriseerimise etapis sätestatud olulised nõuded. Rakendus võimaldab valida kahe sõelumise ja kahe optimeerimise disaini vahel, importida eksperimentidest saadud toorandmed ning teostab ja kuvab toorandmete pealt automaatselt arvutatud tulemused. Arendatud veebiteenus võimaldab keskenduda keemiapoolsele sisule nagu ülesande püstituse eesmärk oli.

Lõputöö käigus sai teostatud nii vajalikud analüüsid lõputöö jaoks vajalikeks arendusteks kui ka kaardistatud tulevikuarenduste võimalused. Tarkvara arendati edasise arendusperspektiive silmas pidades. Tagarakenduse arendamisel kasutati arenduskeelet Pythoni raamistikku FastAPI. Python omab väga suurt valikut teeki statistiliste arvutuste läbiviimiseks ning FastAPI on sobiv kiirelt väiksemate rakenduste ehitamiseks. Klientrakendus arendusel kasutati JavaScripti teeki React, selle suure kommuuni ning sobilikkuse tõttu lihtsate üheleherakenduste arenduseks. Tarkvara evitamisel kasutati juba eelnevalt kasutusel olnud konteineriseeritud mikroteenuste arhitektuuri.

Bakalaureusetöö eesmärgid said täidetud. Esialgsest minimaalsest versioonist sai mahukama versiooni, kui algselt planeeritud, mis võimaldab paremat kaasamist järgmises veebikursuse andmise tsüklis. Mikroteenus on võetud kasutusele ValChrom rakenduse voogudes ning on kättesaadav aadressil <https://valchrom.ut.ee/doi>.

Kasutatud kirjandus

- [1] „MOOC: Validation of liquid chromatography mass spectrometry (LC-MS) methods (analytical chemistry) course“. https://sisu.ut.ee/lcms_method_validation/course-introduction (vaadatud 20. aprill 2023).
- [2] „ValChrom“. <https://valchrom.ut.ee/#/> (vaadatud 20. aprill 2023).
- [3] „10.4 Experimental design“. https://sisu.ut.ee/lcms_method_validation/104-experimental-design?lang=en (vaadatud 15. aprill 2023).
- [4] P. K. Sahu, N. R. Ramiseti, T. Cecchi, S. Swain, C. S. Patro, ja J. Panda, „An overview of experimental designs in HPLC method development and validation“, *Journal of Pharmaceutical and Biomedical Analysis*, kd 147, lk 590–611, jaan 2018, doi: 10.1016/j.jpba.2017.05.006.
- [5] I. Leito, „LC-MS Validation online course 7th run has finished!“, *Chair of Analytical Chemistry*, 15. veebruar 2023. <https://analytical.chem.ut.ee/2023/02/lc-ms-validation-online-course-7th-run-has-finished/> (vaadatud 10. aprill 2023).
- [6] K. Herodes, „ValChrom - validation of chromatographic methods“. <https://sisu.ut.ee/valchrom/main-page> (vaadatud 10. aprill 2023).
- [7] G. A. Okolo, „ValChrom – Software Tool for Validation of Chromatographic Analysis Method“, Thesis, 2019. Vaadatud: 24. aprill 2023. [Online]. Available at: https://comserv.cs.ut.ee/home/files/okolo_softwareengineering_2019.pdf?study=ATILoputoo&reference=827C5EB4FBC7718A06D63B65E60A82FBC85ACF7E
- [8] K. H.-F. Toulassi, „Tarkvaraline tööriist kromatograafilise meetodi valideerimiseks“, Thesis, 2019. Vaadatud: 24. aprill 2023. [Online]. Available at: <https://dspace.ut.ee/handle/10062/66445>
- [9] A. Kruve *et al.*, „Tutorial review on validation of liquid chromatography–mass spectrometry methods: Part II“, *Analytica Chimica Acta*, kd 870, lk 8–28, apr 2015, doi: 10.1016/j.aca.2015.02.016.
- [10] „Ametist | Ravimiamet“. <https://www.ravimiamet.ee/asutus-uudised-kontaktid/ravimiameti-too> (vaadatud 15. aprill 2023).
- [11] „10. Ruggedness, robustness“. https://sisu.ut.ee/lcms_method_validation/10-ruggedness-robustness?lang=en (vaadatud 15. aprill 2023).
- [12] A. Kruve *et al.*, „Tutorial review on validation of liquid chromatography–mass spectrometry methods: Part I“, *Analytica Chimica Acta*, kd 870, lk 29–44, apr 2015, doi: 10.1016/j.aca.2015.02.017.
- [13] T. Tome, N. Žigart, Z. Časar, ja A. Obreza, „Development and Optimization of Liquid Chromatography Analytical Methods by Using AQbD Principles: Overview and Recent Advances“, *Org. Process Res. Dev.*, kd 23, nr 9, lk 1784–1802, sept 2019, doi: 10.1021/acs.oprd.9b00238.

- [14] R. Deidda, S. Orlandini, P. Hubert, ja C. Hubert, „Risk-based approach for method development in pharmaceutical quality control context: A critical review“, *Journal of Pharmaceutical and Biomedical Analysis*, kd 161, lk 110–121, nov 2018, doi: 10.1016/j.jpba.2018.07.050.
- [15] B. Benedetti, V. Caponigro, ja F. Ardini, „Experimental Design Step by Step: A Practical Guide for Beginners“, *Critical Reviews in Analytical Chemistry*, kd 52, nr 5, lk 1015–1028, juuli 2022, doi: 10.1080/10408347.2020.1848517.
- [16] D. B. Hibbert, „Experimental design in chromatography: A tutorial review“, *Journal of Chromatography B*, kd 910, lk 2–13, dets 2012, doi: 10.1016/j.jchromb.2012.01.020.
- [17] I. M. Fukuda, C. F. F. Pinto, C. dos S. Moreira, A. M. Saviano, ja F. R. Lourenço, „Design of Experiments (DoE) applied to Pharmaceutical and Analytical Quality by Design (QbD)“, *Braz. J. Pharm. Sci.*, kd 54, nr spe, nov 2018, doi: 10.1590/s2175-97902018000001006.
- [18] „5. Process Improvement“. <https://www.itl.nist.gov/div898/handbook/pri/pri.htm> (vaadatud 23. aprill 2023).
- [19] S. L. C. Ferreira *et al.*, „Box-Behnken design: An alternative for the optimization of analytical methods“, *Analytica Chimica Acta*, kd 597, nr 2, lk 179–186, aug 2007, doi: 10.1016/j.aca.2007.07.011.
- [20] „Fusion Pro“. https://www.smatrix.com/fusion_pro.html (vaadatud 19. veebruar 2023).
- [21] M. A. Bezerra, R. E. Santelli, E. P. Oliveira, L. S. Villar, ja L. A. Escaleira, „Response surface methodology (RSM) as a tool for optimization in analytical chemistry“, *Talanta*, kd 76, nr 5, lk 965–977, sept 2008, doi: 10.1016/j.talanta.2008.05.019.
- [22] „5.3.3.6.1. Central Composite Designs (CCD)“. <https://www.itl.nist.gov/div898/handbook/pri/section3/pri3361.htm> (vaadatud 22. aprill 2023).
- [23] „5.5.2.1. D-Optimal designs“. <https://www.itl.nist.gov/div898/handbook/pri/section5/pri521.htm> (vaadatud 22. aprill 2023).
- [24] „Microsoft Excel Spreadsheet Software | Microsoft 365“. <https://www.microsoft.com/en/microsoft-365/excel> (vaadatud 20. aprill 2023).
- [25] „MATLAB - MathWorks“. <https://se.mathworks.com/products/matlab.html> (vaadatud 20. aprill 2023).
- [26] „statistica.com by StatSoft (Europe) GmbH: TIBCO Data Science / Statistica™“. <https://www.statistica.com/en/software/tibco-data-science/-/tibco-statistica> (vaadatud 20. aprill 2023).
- [27] „SPSS Statistics | IBM“. <https://www.ibm.com/products/spss-statistics> (vaadatud 20. aprill 2023).
- [28] „Data Analysis Software | Statistical Software Package | Minitab“. <https://www.minitab.com/en-us/products/minitab/> (vaadatud 20. aprill 2023).
- [29] „Data Analysis Software“. https://www.jmp.com/en_ch/software/data-analysis-software.html (vaadatud 20. aprill 2023).

- [30] „MODDE® - Design of Experiments Software“, *Sartorius*, 2021. <https://www.sartorius.com/en/products/process-analytical-technology/data-analytics-software/doe-software/modde> (vaadatud 20. aprill 2023).
- [31] „Fusion Pro - Fusion QbD Software - Quality by Design - DOE - Automated Data Analysis - Chromatography Application Software“. https://www.smatrix.com/fusion_pro.html (vaadatud 20. aprill 2023).
- [32] „OriginLab - Origin and OriginPro - Data Analysis and Graphing Software“. <https://www.originlab.com/index.aspx?> (vaadatud 20. aprill 2023).
- [33] S. T. Inc, „STATGRAPHICS | Data Analysis Solutions“. <https://www.statgraphics.com> (vaadatud 20. aprill 2023).
- [34] „Design-Expert | Stat-Ease“. <https://www.statease.com/software/design-expert/> (vaadatud 20. aprill 2023).
- [35] „R: The R Project for Statistical Computing“. <https://www.r-project.org/> (vaadatud 20. aprill 2023).
- [36] „Welcome to Python.org“, *Python.org*, 13. aprill 2023. <https://www.python.org/> (vaadatud 20. aprill 2023).
- [37] „pyDOE: The experimental design package for python — pyDOE 0.3.6 documentation“. <https://pythonhosted.org/pyDOE/> (vaadatud 19. veebruar 2023).
- [38] „dexpy - Design of Experiments (DOE) in Python — dexpy 0.1 documentation“. <https://statease.github.io/dexpy/> (vaadatud 16. aprill 2023).
- [39] „Welcome to DOEPY — doepy 0.0.1 documentation“. <https://doepy.readthedocs.io/en/latest/> (vaadatud 20. aprill 2023).
- [40] S. Haan, „DoEgen: A Python Library for Optimised Design of Experiment Generation and Evaluation“. 3. aprill 2023. Vaadatud: 16. aprill 2023. [Online]. Available at: <https://github.com/sebhaan/DoEgen>
- [41] R. Sjoegren, „pyDOE2: Design of experiments for Python“. Vaadatud: 16. aprill 2023. [OS Independent]. Available at: <https://github.com/clicumu/pyDOE2>
- [42] A. Business, „Chapter 10: MoSCoW Prioritisation“. <https://www.agilebusiness.org/dsdm-project-framework/moscow-prioritisation.html> (vaadatud 15. aprill 2023).
- [43] „What is an API?“ <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces> (vaadatud 19. aprill 2023).
- [44] „Microservices“, *martinfowler.com*. <https://martinfowler.com/articles/microservices.html> (vaadatud 22. aprill 2023).
- [45] „Micro Frontends“, *martinfowler.com*. <https://martinfowler.com/articles/micro-frontends.html> (vaadatud 22. aprill 2023).
- [46] „Frontend vs Backend“, *GeeksforGeeks*, 11. juuli 2019. <https://www.geeksforgeeks.org/frontend-vs-backend/> (vaadatud 22. aprill 2023).
- [47] U. Groemping, B. Amarov, ja H. Xu, „DoE.base: Full Factorials, Orthogonal Arrays and Base Utilities for DoE Packages“. 9. mai 2022. Vaadatud: 7. aprill 2023. [Online]. Available at: <https://CRAN.R-project.org/package=DoE.base>
- [48] B. Wheeler, „AlgDesign: Algorithmic Experimental Design“. 25. mai 2022. Vaadatud: 7. aprill 2023. [Online]. Available at: <https://CRAN.R-project.org/package=AlgDesign>

- [49] U. Groemping, „FrF2: Fractional Factorial Designs with 2-Level Factors“. 9. mai 2022. Vaadatud: 7. aprill 2023. [Online]. Available at: <https://CRAN.R-project.org/package=FrF2>
- [50] „Shiny“. <https://shiny.rstudio.com/> (vaadatud 20. aprill 2023).
- [51] „Why Spring“, *Why Spring*. <https://spring.io> (vaadatud 22. aprill 2023).
- [52] „What is Java? | IBM“. <https://www.ibm.com/topics/java> (vaadatud 22. aprill 2023).
- [53] „IMSL Java Library | High-ROI Functions for Java Applications“. <https://www.imsl.com/products/imsl-java-libraries> (vaadatud 7. aprill 2023).
- [54] „Math – The Commons Math User Guide - Statistics“. <https://commons.apache.org/proper/commons-math/userguide/stat.html> (vaadatud 7. aprill 2023).
- [55] „The Python Tutorial“, *Python documentation*. <https://docs.python.org/3/tutorial/index.html> (vaadatud 7. aprill 2023).
- [56] S. Goled, „Python’s increasing popularity in scientific and high-performance computing“, *Analytics India Magazine*, 12. aprill 2022. <https://analyticsindiamag.com/pythons-increasing-popularity-in-scientific-and-high-performance-computing/> (vaadatud 15. aprill 2023).
- [57] „Why Python for Scientific Computing“, *PyCodeMates*, 23. veebruar 2023. <https://www.pycodemates.com/2023/02/learning-python-for-scientific-computing.html> (vaadatud 15. aprill 2023).
- [58] „Stack Overflow Developer Survey 2021, Other frameworks and libraries“, *Stack Overflow*. <https://insights.stackoverflow.com/survey/2021#section-most-popular-technologies-other-frameworks-and-libraries> (vaadatud 15. aprill 2023).
- [59] „Python teek: numpy, <https://numpy.org/>“.
- [60] „pandas - Python Data Analysis Library“. <https://pandas.pydata.org/> (vaadatud 20. aprill 2023).
- [61] „SciPy“. <https://scipy.org/> (vaadatud 16. aprill 2023).
- [62] „NumPy“. <https://numpy.org/> (vaadatud 20. aprill 2023).
- [63] „FastAPI“. <https://fastapi.tiangolo.com/> (vaadatud 7. aprill 2023).
- [64] „The OpenAPI Specification“. OpenAPI Initiative, 19. aprill 2023. Vaadatud: 19. aprill 2023. [Online]. Available at: <https://github.com/OAI/OpenAPI-Specification>
- [65] „Flask“, *Pallets*. <https://palletsprojects.com/p/flask/> (vaadatud 7. aprill 2023).
- [66] „Django“, *Django Project*. <https://www.djangoproject.com/> (vaadatud 7. aprill 2023).
- [67] „Introduction — statsmodels“. <https://www.statsmodels.org/dev/index.html> (vaadatud 16. aprill 2023).
- [68] „scikit-learn: machine learning in Python — scikit-learn 1.2.2 documentation“. <https://scikit-learn.org/stable/> (vaadatud 8. aprill 2023).
- [69] „Benefits of Using Frameworks For Web Application Development“, *Oodles ERP*, 6. aprill 2021. <https://erpsolutions.oodles.io/blog/web-development-frameworks-benefits/> (vaadatud 16. aprill 2023).

- [70] 262588213843476, „Front-end frameworks popularity (React, Vue, Angular and Svelte“, *Gist*. <https://gist.github.com/tkrotoff/b1caa4c3a185629299ec234d2314e190> (vaadatud 7. aprill 2023).
- [71] M. Bostock, „D3.js - Data-Driven Documents“. <https://d3js.org/> (vaadatud 7. aprill 2023).
- [72] „Plotly Javascript, vabavaraline teek, <https://plotly.com/javascript/>“.
- [73] „Vue.js - The Progressive JavaScript Framework | Vue.js“. <https://vuejs.org/> (vaadatud 22. aprill 2023).
- [74] „React“. <https://react.dev/> (vaadatud 22. aprill 2023).
- [75] „Angular - Introduction to the Angular docs“. <https://angular.io/docs> (vaadatud 22. aprill 2023).
- [76] „React“. <https://plotly.com/javascript/react/> (vaadatud 14. aprill 2023).
- [77] „React Three Fiber Documentation“, *React Three Fiber Documentation*. <https://docs.pmnd.rs/react-three-fiber> (vaadatud 15. mai 2023).
- [78] „vis.js“. <https://visjs.org/> (vaadatud 15. mai 2023).
- [79] „Papa Parse - Powerful CSV Parser for JavaScript“. <https://www.papaparse.com/> (vaadatud 14. aprill 2023).
- [80] „Getting Started: Overview“. <https://react-table.tanstack.com/docs/overview> (vaadatud 14. aprill 2023).
- [81] „Introducing ChatGPT“. <https://openai.com/blog/chatgpt> (vaadatud 8. aprill 2023).
- [82] „GitHub Copilot · Your AI pair programmer“, *GitHub*. <https://github.com/features/copilot> (vaadatud 8. aprill 2023).
- [83] „ChatGPT tutorial: How to easily improve your coding skills with ChatGPT“, *Lab Lab*. <https://lablab.ai/t/chatgpt-tutorial-how-to-easily-improve-your-coding-skills-with-chatgpt> (vaadatud 15. aprill 2023).
- [84] „Ordinary Least Squares (OLS) using statsmodels“, *GeeksforGeeks*, 9. juuli 2020. <https://www.geeksforgeeks.org/ordinary-least-squares-ols-using-statsmodels/> (vaadatud 15. aprill 2023).
- [85] S. Dixit, „Beginners Guide to Client Server Communication“, *Medium*, 30. juuli 2019. <https://medium.com/@subhangdxt/beginners-guide-to-client-server-communication-8099cf0ac3af> (vaadatud 16. aprill 2023).
- [86] „What is REST API (RESTful API)?“, *App Architecture*. <https://www.techtarget.com/searchapparchitecture/definition/RESTful-API> (vaadatud 16. aprill 2023).
- [87] „What is a REST API?“ <https://www.redhat.com/en/topics/api/what-is-a-rest-api> (vaadatud 16. aprill 2023).
- [88] „What is REST“, *REST API Tutorial*, 7. aprill 2022. <https://restfulapi.net/> (vaadatud 19. aprill 2023).
- [89] „What is RESTful API? - RESTful API Explained - AWS“, *Amazon Web Services, Inc.* <https://aws.amazon.com/what-is/restful-api/> (vaadatud 18. aprill 2023).
- [90] „Best Practices in API Design“. <https://swagger.io/resources/articles/best-practices-in-api-design/> (vaadatud 16. aprill 2023).

- [91] „Component-Based Architecture“. https://www.tutorialspoint.com/software_architecture_design/component_based_architecture.htm (vaadatud 14. aprill 2023).
- [92] „Component Architecture | Hands on React“. <https://handsonreact.com/docs/component-architecture> (vaadatud 14. aprill 2023).
- [93] „Writing Markup with JSX – React“. <https://react.dev/learn/writing-markup-with-jsx> (vaadatud 14. aprill 2023).
- [94] „Create React App“. <https://create-react-app.dev/> (vaadatud 10. aprill 2023).
- [95] „SPA (Single-page application) - MDN Web Docs Glossary: Definitions of Web-related terms | MDN“, 21. veebruar 2023. <https://developer.mozilla.org/en-US/docs/Glossary/SPA> (vaadatud 16. aprill 2023).
- [96] P. Bolmér, „How To Implement a Single Page Application Using React-Router“, *Medium*, 25. november 2022. <https://betterprogramming.pub/how-to-implement-a-single-page-application-using-react-router-cc6b9e5c3aec> (vaadatud 16. aprill 2023).
- [97] M. Poppendieck ja T. Poppendieck, *Lean Software Development: An Agile Toolkit*. Boston: Addison-Wesley Professional, 2003.
- [98] W. L. in R.-B. U. Experience, „Response Time Limits: Article by Jakob Nielsen“, *Nielsen Norman Group*. <https://www.nngroup.com/articles/response-times-3-important-limits/> (vaadatud 8. aprill 2023).
- [99] „Performance is User Experience | Designing for Performance“. <https://designingforperformance.com/performance-is-ux/#designers-impact-on-performance> (vaadatud 14. aprill 2023).
- [100] „First Input Delay (FID)“, *web.dev*. <https://web.dev/fid/> (vaadatud 14. aprill 2023).
- [101] „Time to Interactive“, *Chrome Developers*. <https://developer.chrome.com/docs/lighthouse/performance/interactive/> (vaadatud 16. aprill 2023).
- [102] „Docker Compose overview“, *Docker Documentation*, 14. aprill 2023. <https://docs.docker.com/compose/> (vaadatud 14. aprill 2023).
- [103] „Docker overview“, *Docker Documentation*, 14. aprill 2023. <https://docs.docker.com/get-started/overview/> (vaadatud 15. aprill 2023).
- [104] „Adding Custom Environment Variables | Create React App“, 29. mai 2020. <https://create-react-app.dev/docs/adding-custom-environment-variables/#what-other-env-files-can-be-used> (vaadatud 9. aprill 2023).
- [105] „NGINX Reverse Proxy | NGINX Plus“. <https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/> (vaadatud 15. aprill 2023).
- [106] „How Secure Are Popular Web Frameworks? Here Is a Comparison“, *Veracode*. <https://www.veracode.com/blog/secure-development/how-secure-are-popular-web-frameworks-here-comparison> (vaadatud 16. aprill 2023).
- [107] „OWASP Top Ten | OWASP Foundation“. <https://owasp.org/www-project-top-ten/> (vaadatud 9. aprill 2023).

- [108] D. Gaither, „API Keys: API Authentication Methods & Examples“. <https://blog.stopligh.io/api-keys-best-practices-to-authenticate-apis> (vaadatud 20. aprill 2023).
- [109] Cure53, „DOMPurify“. 9. aprill 2023. Vaadatud: 9. aprill 2023. [Online]. Available at: <https://github.com/cure53/DOMPurify>
- [110] „SlowApi Documentation“. <https://slowapi.readthedocs.io/en/latest/> (vaadatud 23. aprill 2023).

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Asko Laaniste

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Eksperimendi disaini mikroteenus", mille juhendajateks on Kristiina Hakk ja Riin Rebane,
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

15.05.2023

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