

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Vidrik Toom Tabas

Eestikeelne ravimite koostoime kontrollimise veebirakendus

Bakalaureusetöö (9 EAP)

Juhendaja: Lidia Feklistova, PhD

Tartu 2024

Eestikeelne ravimite koostoime kontrollimise veebirakendus

Lühikokkuvõte:

Bakalaureusetöö eesmärk on luua kasutajasõbralik ja kõigile kättesaadav ravimite koostoime kontrollimise eestikeelne veebirakendus. Enne rakenduse loomist tutvus töö autor olemasolevate lahendustega. Veebirakenduse loomisel kasutati erinevaid raamistike ning andmebaasi juhtimissüsteemiks valiti PostgreSQL. Kasutades loodud veebirakendust saavad selle kasutajad vastuse ning kirjelduse ravimite sobivuse kohta. Veebirakendus ning tulemused on eesti keeles, ainsa avaliku alternatiivina samalaadsetele rakendustele, mis on mõeldud kliinilistele asutustele ning on kättesaadavad vaid tervishoiutöötajatele. Andmed pärinevad Norra riiklikust andmebaasist NOMA.

Võtmesõnad:

Veebirakendus, Vue.js, Nuxt.js, JavaScript, Python, PostgreSQL, ravimite koostoime, Ravimiregister

CERCS: P175 Informaatika, süsteemiteooria

Web application in Estonian for checking drug interactions

Abstract:

The bachelor's thesis is an applied thesis with the aim of creating a user friendly and easily accessible web application for checking drug interactions. Before creating the application, the author familiarized themselves with existing solutions. Various frameworks were used for creating the web application, with PostgreSQL as the database management system. The developed web application provides visitors with a response and description of the suitability of the drugs. The web application and results are in Estonian, serving as the only public alternative to similar applications, which are meant for clinical institutions and are accessible only by medical personnel. The data originates from the Norwegian national database NOMA.

Keywords:

Web application, Vue.js, Nuxt.js, JavaScript, Python, PostgreSQL, drug interactions, Ravimiregister

CERCS: P175 Informatics, systems theory

Sisukord

Sissejuhatus	5
1. Mõisted ja terminid	7
2. Ravimite koostoime ja olemasolevad lahendused.....	9
2.1 Ravimite koostoime	9
2.2 Ülevaade olemasolevatest lahendustest	10
3. Veebirakenduse lahenduse nõuded	12
3.1 Funktsionaalsed ja mittefunktsionaalsed nõuded.....	12
3.2 Lahenduse disain ja prototüüp	14
4. Kasutatud tehnoloogiad.....	16
4.1 Andmed ja nende töötlemine	16
4.2 Veebiraamistike võrdlus.....	16
4.2.1 React.....	17
4.2.2 Vue.js	18
4.2.3 Angular.....	19
4.3 Veebirakenduse avalikustamine.....	19
4.4 Kasutatud tehnoloogiate valik.....	20
5. Lahendus	22
5.1 Andmed.....	22
5.2 Andmete töötlemine	23
5.3 Veebirakendus.....	25
5.3.1 Veebirakenduse vaated.....	26
5.3.2 Veebirakenduse avalikustamine	29
6. Lahenduse analüüs ning kasutajasõbralikkuse testimine	30
6.1 Vastavus nõuetele.....	30
6.2 Kasutajasõbralikkuse testimise	32

6.2.1 Kasutajasõbralikkuse testimise tulemused	33
6.2.2 Vastajate kommentaarid	34
7. Edasised arenguvõimalused	35
Kokkuvõte	36
Viidatud kirjandus	37
Lisad	40
I. Vahelehe „Abi“ sisu	40
II. Litsents	41

Sissejuhatus

Ravimid ja nende areng on inimeste elukvaliteeti märkimisväärselt tõstnud ning see toimub jätkuvalt. Kuid siiski on ravimite võtmisel ka negatiivseid külgi. Üheks nendest on ravimite omavaheline koostoime. See koostoime võib tuua kaasa väga ebameeldivaid ja ohtlikke tagajärgi, nagu toimeainete võimendumine või nõrgenemine, kõrvaltoimete suurenenud riski ning ravimite toime muutust organismis.

Kahjuks ravimite tarvitajaid ei tea või neil pole ligipääsu allikatele, kust saada teada mis ravimeid nad võivad koos kasutada ning millised omavahelised koostoimed võivad esineda. Ravimite koostoimete teadmatuse probleemi märkas käesoleva töö autor ka enda ja lähedaste juures. Kuid proovides selle kohta rohkem infot saada, ei leidnud töö autor Eesti maastikult piisavalt infot ja meditsiini valdkonnaga mitteseotud inimestele vabalt kättesaadavat veebirakendust vms. Ilma tervishoiutöötajatelt küsimata, mis olenevalt olukorrast ja ajast ei ole alati võimalik, jääb valikuks vaid ravimite pakendi infolehed. Kuid vajaliku info välja lugemine ja erialaste teadmiste puudumise tõttu võib olla raskendatud ja aeganõudvam. Kliinilistele asutustele ja seal töötavatele tervishoiutöötajatele on Eestis kasutusel ettevõtte Synbase¹ välja töötatud andmebaas ja rakendus. On ka olemas erinevad välismaised andmebaasid ja veebirakendused, kuid neist ükski ei võimalda eestikeelset kasutamist ja otsingutulemuste väljastamist.

Bakalaureusetöö on rakenduslik töö ning selle eesmärk on luua kasutajasõbralik ja kõigile kättesaadav ravimite koostoime kontrollimise veebirakendus. Selle kasutamise tulemusena peavad külastajad saama eestikeelse vastuse ning eestikeelse kirjelduse ravimite sobivuse kohta. Veebirakenduse andmete saamiseks kasutatakse Norra riikliku meditsiinivahendite ameti NOMA (*Norwegian Medical Products Agency*) andmebaasi². Vajaliku ravimite koostoimete info saamiseks andmebaasist kasutatakse programmeerimiskeeli Python ja JavaScript, lugedes andmed PostgreSQL andmebaasi. Veebirakendus ehitatakse veebiraamistiku Vue.js ning selle alamraamistiku Nuxt.js peale.

Antud bakalaureusetöö koosneb seitsmest sisulisest peatükist. Esimeses peatükis tuuakse välja töös kasutatavad mõisted ja terminid. Teises peatükis tutvustatakse lähemalt erinevate ravimite koostoimet ja selle ohtlikkust ning kuidas seda kontrollida, kus antakse ka ülevaade

¹ <https://app.synbase.eu/en/product/drug-interactions-database/>

² <https://www.dmp.no/om-oss/distribusjon-av-legemiddeldata>

olemasolevatest lahendustest. Kolmandas peatükis tuuakse välja töö funktsionaalsed ja mittefunktsionaalsed nõuded ning veebirakenduse disaini ja prototüübi loomine. Neljandas peatükis loetletakse töö tegemiseks võimalikud tehnoloogiad, võrreldakse nende plusse ja miinuseid ning valitakse tööks sobivaimad. Viiendas peatükis kirjeldatakse veebirakenduse loomise protsessi, tekkinud probleeme ja valminud lahendust. Kuuendas peatükis analüüsitakse valminud lahendust, kontrollitakse valminud veebirakenduse vastavust nõuetele ning kirjeldatakse kasutajasõbralikkuse testimist. Seitsmendas peatükis tuuakse välja võimalikud edasiarengu suunad.

1. Mõisted ja terminid

Andmebaas (ingl *database*): infokogum, mis on organiseeritud sellisel viisil, et arvutiprogramm suudab sellest kiiresti üles leida vajalikud andmed [1].

Andmebaasi juhtimissüsteem, ehk **andmebaasihaldur** (ingl *database management system*, DBMS): tarkvarasüsteem andmebaaside loomiseks, haldamiseks ja kasutamiseks [1].

Avatud lähtekood (ingl *open source*): mistahes tarkvara, mille lähtekood on tehtud programmeerijatele ja kasutajatele kättesaadavaks nii kasutamiseks kui muutmiseks [1].

Funktsionaalsed nõuded (ingl *functional requirements*): süsteemi käitumise kirjeldus konkreetsetes tingimustes [2].

Käsureainterpretaator, ehk **käsuprotsessor** (ingl *command line interpreter*, CLI): operatsioonisüsteemi sisse kuuluv programm, mis võtab vastu ja täidab operatsioonisüsteemi käske [1].

Mittefunktsionaalsed nõuded (ingl *non-functional requirements*): süsteemi omaduse või tunnuse kirjeldus, mida see peab näitama või piirang, mida see peab järgima [2].

Prototüüp (ingl *prototype*): esialgne teostus või mudel lahenduse, näitajate ja võimaluste hindamiseks või nõuete selgitamiseks, täpsustamiseks [3].

Ravimi kõrvaltoime (ingl *adverse drug reaction*, ADR): kahjulik ning soovimatu reaktsioon ravimile, mis tekib haiguse diagnoosimise, profülaktika või ravi käigus ravimi tavaliste annuste kasutamisel [4].

Ravimite koostoime (ingl *drug-drug interaction*, DDI): ravimite vaheline reaktsioon, kus erinevad ravimid või nende kombinatsioonid võivad mõjutada üksteise toimet organismis [5].

Server (ingl *server*): programm, mis pakub teenuseid teistele programmidele (klientprogrammidele) [1].

Sisuedastusvõrk (ingl *content delivery network*, CDN): internetis või suures intranetis laiali paigutatud sisuga süsteem, kus sisust tekitatakse palju koopiaid ja puhverdatakse neid üle kogu võrgu [1].

Töötleja (ingl *handler*): alamprogramm, mis täidab teatud kindlat ülesannet [1].

Vahemälu (ingl *cache, cache-memory*): mälu sageli kasutatavate andmete ajutiseks hoidmiseks [1].

Veebiraamistik (ingl *web application framework*): struktureeritud kogum teeke ja tööriistu veebirakenduste loomiseks, hõlmates komponente nagu HTTP päringute haldamine, marsruutide käsitus, side andmebaasidega ja kasutajaliideste vaadete kuvamine [6].

Veebirakendus (ingl *web application*): veebis paiknev tarkvara [1].

2. Ravimite koostoime ja olemasolevad lahendused

Antud peatükis kirjeldatakse ravimite koostoime olemust ning selle ohtu organismile. Tuuakse välja ka Eestis ja mujal maailmas läbiviidud uuringud ravimite tarbimise ning ravimite koos- ja kõrvalmõjude esinemise kohta. Samuti antakse ülevaade olemasolevatest ravimi koostoimete kontrollimise lahendustest, nende ligipääsust, kasulikkusest ning nende plussidest-miinustest.

2.1 Ravimite koostoime

Ükski ravim pole lõplikult ohutu. Igal toimeainel on organismile mõju ning seetõttu on väga oluline ravimite õige kasutamine. Ravimite koostoime on erinevate toimeainete omavaheline vastastikmõju. Mitmete erinevate ravimite samaaegsel kasutamisel suureneb tõenäosus koos- ja kõrvaltoimete esinemisele. Need koostoimed võivad põhjustada mitmesuguseid soovimatuid tagajärgi, sealhulgas ühe või mitme ravimi terapeutiliste mõjude võimendumist või nõrgenemist, kõrvaltoimete suurenenud riski või ravimite toime muutust organismis. Ravimite tarvitajad peavad olema valvsad potentsiaalsete koostoimete tuvastamisel ja ettevaatusabinõude rakendamisel, et saavutada soovitud tulemused ravimite võtmisel ning tagada ohutus [7].

Aastal 2023 Eestis läbi viidud uuringu põhjal leiti, et 82% Eesti elanikest tarvitavad retseptiravimeid ja see arv tõuseb iga aastaga [8]. See on vaid retseptiravimite kohta, mille kõrvalt tarvitatakse ka käsimüügiravimeid ning looduslike ravimeetodeid. Lisaks on ülemaailmselt tõusvas joones polüfarmakoteraapia (ingl *polypharmacy*), mis tähendab mitme, tavaliselt viie või enama, ravimi samaaegset üleliigset tarvitamist [9]. Siinkohal saab veel rõhutada, et mida rohkem toimeaineid on organismis, seda suurem võib olla negatiivsete koostoimete risk.

Arenenud riikides on ravimite koos- ning kõrvaltoimed on üks suurimatest kliinilistest põhjustest, millega haiglasse satutakse ning millega tervishoiutöötajad peavad tegelema [10]. Ka Eestis on tehtud prospektiivne vaatlusuuring – ravimite koos- ja kõrvaltoimed EMO-sse³ pöördumise põhjusena [11]. Uuringust leiti, et kõikidest uuringus osalenutest 9% ning vaatlusperioodil EMO-sse pöördunutest 6,4%, moodustasid ravimi võimaliku kõrval- või koostoimega seotud patsiendid. Sellist uuringut ei ole Eestis varem tehtud, seega

³ Erakorralise meditsiini osakond

ei saa varasemate tulemustega otseselt võrrelda, kuid selle eest saab võrdluse allikaks võtta ülejäänud maailma. Näiteks toob Eesti uuring ka välja teiste sarnaste uuringute tulemused, kus koos- ja kõrvaltoimetega haiglasse pöördumiste põhjustest moodustas Rootsis 12%, Taanis 10,6%, Suurbritannias 6,5% ning USA-s 6,7%. Uuringu autorite arvamusel erinevad tulemused võivad olla tingitud mitmest asjaolust, nagu näiteks suurem uuritavate hulk, kuid üldjoontes on need numbrid väga sarnased enamustes arenenud riikides.

Ravimite koostoimet saab klassifitseerida mitut erinevat moodi [12], kuid antud töös on need jagatud kolmeks – kõrge, mõõduka ning minimaalse kliinilise tähtsusega. Klassifikatsiooni kategooriad on valitud vastavalt antud töös kasutatavatele andmetele, kus iga ravimite koostoime on määratud kategooriaga "Tuleks vältida", "Tuleks rakendada ettevaatusabinõusid" või "Tegevust ei ole vaja (koostoime puudub)" [13]. Kõrge kliinilise tähtsusega kaasneb suur koostoime risk, kus tervishoiutöötaja peaks patsiendi olukorda regulaarselt jälgima, vajadusel kaaluma alternatiivset ravimit. Mõõduka kliinilise tähtsuse korral peaks võimalusel hoiduma kombinatsioonidest, kasutades neid ainult erijuhtudel ning tervishoiutöötaja jälgimine on tungivalt soovitatav. Minimaalne kliiniline tähtsus tähendab, et koostoime on ebatõenäoline, väike, ebaoluline või ei ole selle koostoime kohta piisavalt informatsiooni.

2.2 Ülevaade olemasolevatest lahendustest

Eelnevalt väljatoodu pärast on äärmiselt oluline, et ravimite tarvitajad teaksid, mis ravimite koostoimed võivad neile ohtlikud olla. Kuid Eestis ei ole selleks avalikult kättesaadavat allikat. Järgnevalt on välja toodud mõned võimalused ja olemasolevad lahendused.

Esimese abi võimalusena on muidugi tervishoiutöötajad. Kohe apteegist ravimeid soetades, on ravimite tarvitajatel võimalik küsida ravimitega seotud koostoimeid. Patsiendile retseptiravimeid müües on apteeker isegi kohustatud tooma välja kõik koostoimed, mis nende vahel võivad esineda. Kuid see ei hõlma enda all käsimüügiravimeid, nende kohta peab patsient ise abi küsima. Võimalus on ka oma perearstilt või -õelt sellelaadset infot saada, kuid see pole igal ajahetkel võimalik. Lisaks on tervishoiutöötajate koormus praeguselt väga suur [14]. Õnneks on olukorra lahendamiseks järjest enam arendusi, näiteks hiljuti arendatud <https://www.minudoc.ee/> veebiplatvorm. Lehel kliente nõustava apteegi proviisori Margot Lehari sõnul on üks sagedasemaid küsimusi, millega apteekri e-konsultatsiooni pöörduakse, seotud ravimite koostoime ja omavahelise sobivusega, kuna tihtipeale tarbivad inimesed korraga mitut ravimit [15].

Teisena on iga ravimiga kaasas pakendi infolehed, millelt saab välja lugeda selle kindla ravimiga võimalikud koostoimed. Kui ravimitarvitajal pole enam hetkel füüsilisele paberile ligipääsu, saab pakendi infolehed kätte ka Ravimiregistri⁴ veebilehe otsingust, otsides kindla pakendi, toimeaine või ATC koodi⁵ järgi. Tulemused võivad eestikeelsele vastele sisaldada ka vene- ja ingliskeelseid pakendi infolehti. Kuid selliselt lehelt koostoimete välja lugemine pole mugav ja kiire. Suurest loetelust peab otsima välja ravimite koostoime, mis teiste võetavate ravimitega esineb või mitte.

Kolmandaks valikuks on ettevõtte Synbase poolt 2016. aastal valminud ravimite koostoimete kontrollimise andmebaas ja kliiniline otsusetugi [16]. Andmebaas on väga kasulik arstidele, õdedele ja apteekritele. Rakendus annab koheselt potentsiaalsete ravimite koostoimete kohta tulemused erinevates kategooriates, jälgides patsiendile varem välja kirjutatud retseptiravimeid. Praeguseks on Synbase rakendus kasutusel enamustes Eesti kliinilistes ettevõtetes, tavakodanikud sellele ligi ei pääse.

Neljandana on valikus võõrkeelsed ravimite koostoimete kontrollimise (ingl *drugs interaction checker*) andmebaasid ja veebirakendused, nagu näiteks WebMD⁶, Drugs⁷, Medscape⁸, DrugBank⁹ ja paljud teised. Kuid ükski neist lehtedest ei ole eestikeelne, ega ka eestikeelsete otsingutulemustega. Ning lisaks tihti taolised välismaised andmebaasid, enamusest pärit Ameerika Ühendriikidest, ei sisalda kõiki Eesti või Euroopa ravimeid.

Arvestades eespool toodud aspekte otsustas töö autor luua rakenduse, mis on kõigile kättesaadav ning annab esmase vastuse ravimite koostoimete kohta. Veebirakendus ning tulemused on täielikult eestikeelsed. Otsitud ravimite tulemuse põhjal annab rakendus ravimi tarvitajale ülevaate ravimite koostoimete esinemisest ning ohtlikkusest, suunates edasiste küsimuste korral professionaalse meditsiinilise abi suunas.

⁴ <https://www.ravimiregister.ee>

⁵ Anatoomilis-terapeutiline keemiline (ingl *anatomical therapeutic chemical*) kood

⁶ <https://www.webmd.com/interaction-checker>

⁷ https://www.drugs.com/drug_interactions.html

⁸ <https://reference.medscape.com/drug-interactionchecker>

⁹ <https://go.drugbank.com/drug-interaction-checker>

3. Veebirakenduse lahenduse nõuded

Antud peatükis on toodud ülevaade funktsionaalsetest ja mittefunktsionaalsetest nõuetest loodava veebirakenduse vastu. Samuti on kirjeldatud veebipõhise rakenduse disaini ja prototüübi loomist.

3.1 Funktsionaalsed ja mittefunktsionaalsed nõuded

Järgnevates tabelites Tabel 1 ja Tabel 2 on esitatud lahenduse tähtsamaid nõudeid. Esimeses tabelis on lahenduse funktsionaalsed nõuded, mis keskenduvad funktsionaalsuse kirjeldamisele ehk mida täpselt lahendus teeb või kuidas käitub ning teises tabelis on lahenduse mittefunktsionaalsed nõuded, ehk on lahenduse funktsionaalsust täpsustavad nõuded, mis aitavad hinnata lahenduse käitumist [2]. Funktsionaalsed ja mittefunktsionaalsed nõuded on iga rakenduse juures väga olulised, kuna nende puudumisel võib lahendus tulla ebakorrektna [17].

Tabel 1. Funktsionaalsed nõuded.

Nõude kood	Nõue
REQ 1	Ravimite koostoime kontrollimise veebirakendus peab andma kasutajale infot veebilehe olemusest, otstarbest ning lahendatavast probleemist ja/või võimalust seda infot saada teistel lehtedel/välistel linkidel.
REQ 2	Veebirakenduse päises peab olema link avalehele kirjaga „Avaleht“.
REQ 3	Veebirakenduse päises peab olema link „Abi“ lehele.
REQ 4	Veebirakenduse päises peab olema link „Otsing“ lehele.
REQ 5	Veebirakendus peab kuvama veateatest informeerivat lehte, kui peaks esinema rakenduse tööd lõhkuv kriitiline veateade. Sama lehte peab ka kuvama mitte eksisteerivaid aadresse külastades.
REQ 6	Veebirakenduse avalehel peab olema lehte iseloomustav pealkiri, mis teistel lehtedel peab asetsema väiksemalt lehe päises.
REQ 7	Veebirakendus ja otsingu tulemused peavad olema eesti keeles.
REQ 8	Veebirakenduse avalehe keskel ja teiste lehtede jaluses peab olema hoiatus, et see veebirakendus ei ole asenduseks päris meditsiinilise personali abi ja info asemel.
REQ 9	Vahelehel "Abi" on välja toodud infot rakenduse loomusest ning taustast, informatiivset teksti ravimite koostoimest ning abistavaid viiteid välistele linkidele.

REQ 10	Veebirakendus peab kuvama tulemusi kasutusel olevate ravimite kohta ning tulemuste puudumisel kasutajat sellest teavitama.
REQ 11	Veebirakenduse kasutaja peab saama otsinguvormi sisestada vähemalt kaks ravimi toimeainet ja/või ATC koodi.
REQ 12	Veebirakenduse otsingu sektsioon peab ebasobilike otsingu terminite korral andma sellest kasutajale märku tagastades „Ravimit ei leitud“.
REQ 13	Veebirakenduse otsingu väljad peavad kasutajale pakkuma sisestatule sobivaid vasteid, mis aitaks ja kiirendaks otsingut.
REQ 14	Veebirakenduse otsingu sektsioon peaks esmaselt sisaldama kahte täidetavat välja, ning nende kõrval olevat nuppu, mis väljasid juurde lisaks. Samuti peab olema nupp lisatud väljade eemaldamiseks.
REQ 15	Sõltuvalt otsingutulemustest kasutajale peab olema kuvatud ravimite koostoime tähtsuse kategooria „Tuleks vältida“, „Tuleks rakendada ettevaatusabinõusid“ või „Kliiniliselt ohutu koostoime“.
REQ 16	Tulemused peavad näitama ravimite koostoime kliinilise tagajärge kirjeldust.
REQ 17	Tulemused peavad näitama ravimite koostoime olukorra kriteeriumit, kui see eksisteerib.
REQ 18	Tulemused peavad näitama ravimite koostoime kohta abinõud, kui see eksisteerib.
REQ 19	Tulemused peavad näitama Ravimiregistri ravimi pakendi infolehe koostoimete sektsiooni, kui see eksisteerib. Lisaks viitav link ravimi pakendi infolehele.

Tabel 2. Mittefunktsionaalsed nõuded.

Nõude kood	Nõue
REQ 20	Ravimite koostoime kontrollimise veebirakenduse väljanägemine ja kasutamine peavad olema tavakasutajale koheselt selgelt mõistetavad. See hõlmab selget navigeerimist ja intuitiivset kasutajaliidese kujundust.
REQ 21	Veebirakendus peab olema heleda värviskeemiga, lisades värvi siniste toonidega meditsiinilistele lehtedele omaselt.
REQ 22	Veebirakendus peab olema internetist kättesaadav 24/7.
REQ 23	Algne veebirakendus ei tohi laadida üle kolme sekundi, kusjuures otsingu päring ei tohi laadida üle kümne sekundi.

REQ 24	Veebirakendus peab suutma hallata suurenenud kasutajate koormust, ehk vähemalt 10 samaaegset kasutajat, ning suuremat andmemahutust ilma olulise jõudluse halvenemiseta ning hangumiseta.
REQ 25	Veebirakendus peab iga kasutaja tegevusele visuaalselt reageerima.
REQ 26	Veebirakenduse Google Lighthouse ¹⁰ skoor peab olema üle 90-e kõikides kategooriates. Seda mõlema ekraani suurusega, <i>desktop</i> ning <i>mobile</i> .
REQ 27	Veebirakendus peab olema mobiilisõbralik.
REQ 28	Veebirakendus peab ühilduma veebibrauseritega Google Chrome, Mozilla Firefox, Safari ja Microsoft Edge ning nende hiljutisemate versioonidega.

3.2 Lahenduse disain ja prototüüp

Vastavalt paika pandud nõuetele oli võimalik luua esmane disain. Lähtudes töö loomusest, milleks on kasutajatele abi ja info väljastamine, prooviti disain luua minimalistlikult. Tänu minimalistliku disaini lähenemisele näevad veebirakendused välja kasutajasõbralikumad ja selgemad, samas suunates kasutajate tähelepanu olulisele sisule [18]. Liigseid elemente eemaldades, tõstavad minimalistlikud veebirakendused kasutaja kaasatust ja arusaamist (mis arendatava veebirakenduse puhul tähendab ravimite koostoime ohtlikkuse rõhutamist) ja kaasatust, kattes efektiivse suhtluse eesmärgi. See lähenemine annab veebirakendustele ka elegantsi, soodustades sujuvat kasutajakogemust.

Disaini loomiseks kasutati Figma¹¹ keskkonda, mis on laialt levinud koostöö disainitööriist. Seda kasutatakse kasutajaliideste, prototüüpide ja interaktiivsete disainide loomiseks reaalajas. Algse disaini eesmärgiks oli veebirakenduse elementide asetus, väljanägemine ning kasutajakogemuse katsetamine.

Esimesena valmis avalehe disain, koos komponentidega mida ka teistel lehtedel kasutatakse. Päisest saab selgelt liikuda vahelehtedele „Abi” ja „Otsing”. Vahelehele „Otsing” on pandud suurem rõhk, sest see on veebirakenduse peamine otstarve. Avalehel on suurelt välja toodud lehte iseloomustav pealkiri põhitoonis. Teisena on pilku püüdvalt punasesse kasti asetatud hoiatus, nagu seda nõuab REQ 8. Avalehe keskel on olemas otsingu sektsioon, mille idee on kasutaja navigeerida „Otsing” vahelehele juba täidetud otsingu terminitega. Vahelehel

¹⁰ <https://developer.chrome.com/docs/lighthouse>

¹¹ <https://www.figma.com/>

“Otsing” korduvad enamus elementidest, lisades otsingu tulemuste sektsiooni. Esinenud ravimite koostoimete vastuse korral näidatakse kasutajale esinevaid väljasid REQ 15 – REQ 19 nõuete kohaselt. Siin on suurem rõhk pandud kliinilise tähtsuse kategooriale, mis näitab vastavates värvitoonides ravimite koostoime esinemist ja ohtu. Nõude REQ 9 kohaselt on vahelehel „Abi“ infot rakenduse loomusest ning taustast, informatiivset teksti ravimite koostoimest ning abistavaid viiteid väliste linkidele.

Kuna Figma abil saab ka väga mugavalt teha prototüüpi, siis valmis ka see samas keskkonnas. Lisades elementidele loogikad, kuhu vastavad interaktsioonid viima peaksid, tekkis pilt sellest, milline päris veebirakendus välja hakkab nägema. Lahenduse prototüüp on kättesaadav aadressilt <https://www.figma.com/proto/4dr13tZPkF5x15MoPoGdX7>.

4. Kasutatud tehnoloogiad

Antud peatükis antakse ülevaade tehnoloogilistest lahenduskäikudest. Loodud veebiarenduse loomise protsessi saab jagada kolmeks etapiks, kus esimene on ravimite koostoitmete andmete leidmine ja haldus, teine veebirakenduse ehitamine lokaalselt ning kolmas veebirakenduse avalikustamine (ingl *deploy*). Nende etappide läbimiseks on palju erinevaid lähenemisviise ja tehnoloogiaid, mida järgnevates alapeatükkides põhjalikumalt kirjeldatakse ja lõplik valik tehakse.

4.1 Andmed ja nende töötlemine

Esimese etapina on vaja panna paika andmete leidmine, nende pärimine ning vajalikuks vormiks töötlemine. Ideaalis oleks parimaks valikuks otsene rakendusliides, ehk API (ingl *Application Programming Interface*) ühendus. Sellisel juhul pärides ühe või mitme toimeaine nimetusega, saadetakse vastu tulemus nende ravimite koostoitmete kohta. Teise valikuna on andmebaas, kust kas algselt või perioodiliselt on võimalik kõik andmed alla laadida või oma andmebaasi edastada. Sellega võib kaasneda suurem ajakulu ja suurenenud ressursid ning ka selle loogika ülesehitus on keerulisem.

Kui andmed on mingil valitud viisil käes, siis enamasti on neid vaja rakendusele sobivaks töödelda. API ühenduse korral tähendaks see vaid valitud väljade filtreerimist ja nende edastamist rakendusele. Kuid andmebaasiga kaasneb terve suure hulga andmete läbi käimist ning vajaliku valimist, mis olenevalt andmebaasi suurusest ja keerukusest võib olla väga pikk ja nõudlik protsess. Töötlemiseks on vajalikud erinevad programmid, skriptid ja keeled, mis töötlevad andmeid ning väljastavad soovitu. Selle protsessi käigus saab ka veel lisada erinevaid vahe funktsioone, nagu näiteks tõlkimine, vormindamine jms, mis andmed sobivale kujule seaksid. Andmebaasi miinuseks on ka see, et seda peab kuskil majutama (ingl *hosting*), mis võib osutuda päris suureks lisakulaks.

4.2 Veebiraamistike võrdlus

Teiseks on erinevate veebiraamistike võrdlus, millega veebirakendus luua. Viimaste aastate populaarseimad veebiraamistikud on püsivalt olnud React, Vue.js ning Angular [19]. Seega järgnevad alapeatükid annavad nendest veebiraamistikest lühikese ülevaate, tuues välja erinevused üksteiste vahel. Iga veebiraamistiku kohta on toodud välja tugevuste ja nõrkuste tabel, mille erinevad punktid põhinevad selles alapeatükis viidatud materjalide kogumiku põhjal ning osaliselt ka autori arvamusel.

4.2.1 React

React on avatud lähtekoodiga JavaScript-i kasutajaliidese teek (ingl *library*), mille arendas välja Meta ning mis on laialdaselt kiidetud oma jõudluse ja paindlikkuse poolest [20]. Erinevalt Angular-ist, mis on põhjalik veebiraamistik, toimib React kasutajaliidese teegina, keskendudes kasutajaliideste loomisele korduvkasutatavate komponentide abil. React-i virtuaalne DOM (ingl *Document Object Model*) võimaldab tõhusaid värskendusi kasutajaliideses, minimeerides otsest manipuleerimist tegeliku DOM-iga ning seeläbi parandades jõudlust [21]. React-is kasutatav süntaksi laiendus JSX (ingl *Javascript Extensible Markup Language*) hõlbustab kasutajaliideste komponentide loomist JavaScript-i koodisisiselt, parandades koodi loetavust ja hooldatavust. React-i ökosüsteem on lai, hõlmates hulgaliselt kolmanda osapoole teeke ja tööriistu, mis täiendavad selle põhifunktsioone [22]. Lisaks sellele, suudab React luua ka mobiilirakendusi sama koodipõhja abil tänu React Native-ile, tõstes veelgi selle atraktiivsust arendajate seas. Kokkuvõtlikult on React-il, kui veebiraamistikul, omad tugevused ning nõrkused, mis on välja toodud Tabel 3.

Tabel 3. React-i tugevused ning nõrkused.

Tugevused	Nõrkused
Jõudlus ja paindlikkus	Pidevad uuendused nõuavad pidevat hooldust
Virtuaalne DOM efektiivsete kasutajaliidese uuenduste jaoks	Lisa õppimiskurv JSX süntaksi tõttu
JSX koodi loetavuse ja arenduste jaoks	Natukene liiga laiali valgunud ökosüsteem
Ulatuslik ökosüsteem paljude kolmanda osapoole teekide ja tööriistadega	
Võimalus luua veebi- ja mobiilirakendusi React Native-iga	
Korduvkasutatavad komponendid.	
Lihtsam õppimiskurv	

4.2.2 Vue.js

Aastal 2014 avalikustatud veebiraamistik Vue.js, kirjeldatud kui progressiivne veebiraamistik, on saanud märkimisväärset tähelepanu oma lihtsuse ja algajasõbralikkuse tõttu [20]. Erinevalt React-ist ja Angular-ist tajutakse Vue.js-i sageli ligipääsetavamana tänu selle vähem järsumale õppimiskurvile. Vue.js on avatud lähtekoodiga ning piiranguteta kasutatav. Kui React on kasutajaliidese teek ja Angular on täisväärtuslik kasutajaliidese raamistik, siis Vue.js saavutab tasakaalu nende kahe vahel, pakkudes raamistikku, mida on lihtne õppida, kuid samas ka võimsust kaasaegsete veebirakenduste loomiseks [23]. Vue.js kasutab reaktiivset (ingl *reactive*) andmesidet, ehk andmete või oleku muutustega värskendatakse DOM-il nende esinemine koheselt [21]. Lisaks on Vue.js välja- ja edasiarendajate prioriteediks lihtsuse tagamine oma süntaksis ja struktuuris, muutes selle populaarseks valikuks erineva oskustasemega arendajate hulgas [22]. Kokkuvõtlikult on Vue.js-il, kui veebiraamistikul, omad tugevused ning nõrkused, mis on välja toodud Tabel 4.

Tabel 4. Vue.js-i tugevused ning nõrkused.

Tugevused	Nõrkused
Lihtsus ja algajasõbralikkus	Väiksem ökosüsteem
Reaktiivne andmeside efektiivsete DOM-i uuenduste jaoks	Piiratud usaldus võrreldes raamistikega, mida toetavad suur korporatsioonid nagu Google ja Facebook
Paindlik ökosüsteem ja modulaarne (ingl <i>modular</i>) arhitektuur	Vähem kaasas olevaid arenenud funktsioone
Tasakaalustatud lähenemine lihtsuse ja võimsuse vahel	Vajalik teekide kasutamine mõningate põhifunktsionaalsuste jaoks
Lihtne õppimiskurv	
Skaleeritavus keskmise suurusega projektides	
Põhjalik dokumentatsioon	

4.2.3 Angular

Suurkorporatsiooni Google poolt arendatav Angular eristub teistest täisfunktsionaalse veebirakenduste raamistikuna, mis põhineb TypeScript-il, ehk tüübipõhisel JavaScript-il [20]. Sarnaselt React-ile ja Vue.js-ile, on Angular samuti avatud lähtekoodiga. Erinevalt Vue.js-ist, mis rõhutab lihtsust, pakub Angular põhjalikku lahendust suurte rakenduste loomiseks [23]. Angular-i ulatuslik funktsioonide komplekt teeb selle sobivaks suurte ettevõtte taseme projektide jaoks [21]. Siiski on Angular-il võrreldes Vue.js-i ja React-iga järsem õppimiskurv, kuna see eeldab TypeScript-i oskust. Angular-il on põhjalik dokumentatsioon ja tugi Google poolt, mis kinnistab arendajate seas usaldust pikaajaliseks hoolduseks ja skaleeritavuseks. Kokkuvõtlikult on Angular-il, kui veebiraamistikul, omad tugevused ning nõrkused, mis on välja toodud Tabel 5.

Tabel 5. Angular-i tugevused ning nõrkused.

Tugevused	Nõrkused
Täisväärtuslik veebirakenduste raamistik	Järsem õppimiskurv
Ulatuslik funktsioonide komplekt	Nõuab TypeScript-i oskust või õppimist
Tugev tugi ja hooldus suurkorporatsioonilt Google	Komplekteeritud rakenduse suurem andmemah
Usaldusväarsus tänu Google-i toetusele	Lihtsamate veebirakenduste jaoks tihti liiga üle arendatud ja komplikeeritud
Skaleeritavus suurte rakenduste puhul	
Järjepidev ja loetav koodibaas	

4.3 Veebirakenduse avalikustamine

Kui rakendus on lokaalses keskkonnas üles seatud, siis kolmandaks sammuks on selle avalikustamine. See tähendab, et rakendus peab jooksmas mingis serveris, millele veebiaadressi kaudu kõikjalt maailmast ligi saab. See server võib joosta ka oma arvutis, kuid enamasti ei ole see soovitatav, sest sel juhul peab see arvuti töötama samal ajal kui lehte külastatakse ning selle koormus võib liiga suureks osutada. Seepärast tavapäraselt kasutatakse selle asemel erinevaid platvorme kes seda teenust pakuvad, mis enamusele on üksteistele sarnased.

Enne rakenduse avalikustamist on vaja ka valida, millisel viisil veebirakenduse sisu kasutajani jõuab. Serveri poolel renderdamine, ehk SSR (ingl *Server-Side Rendering*), ning staatilise saidi genereerimine, ehk SSG (ingl *Static Site Generation*), on kaks levinud lähenemist dünaamilise veebisisu loomiseks [24]. SSR hõlmab iga lehe HTML-i (ingl *HyperText Markup Language*) genereerimist serveri poolel ja selle saatmist kliendile (ingl *client*). See tähendab, et iga kord kui kasutaja lehte külastab, loob server dünaamiliselt HTML-i sisu, kaasates vajaliku andmete hankimise või töötlemise. Vastupidiselt sellele loob SSG HTML-i sisu juba rakenduse ehitamise (ingl *build*) ajal ning saadab selle otse kliendile päringu korral, ilma et neid dünaamiliselt serveris genereeritaks [25]. See tagab kiiremad lehe laadimisajad ja vähendab serveri koormust, kuna sisu on juba ette valmistatud ja selle saab panna sisuedastusvõrgu vahemällu. Siiski pakub SSR rohkem paindlikkust dünaamilise sisu jaoks, mis võib sageli muutuda rakenduse tööajal. Valik SSR-i ja SSG vahel sõltub veebirakenduse konkreetsetest nõuetest, sealhulgas teguritest nagu jõudlus, skaleeritavus ning kui dünaamiliselt uuendatud lehe sisu peab olema.

4.4 Kasutatud tehnoloogiate valik

Tuginedes eespool kirjeldatud tehnoloogiatele, nende eelistele ja puudujääkidele, otsustas töö autor kasutada töös järgmised lahendused.

Andmete pärimiseks oleks parim kasutada API ühendust, millega koheselt veebirakenduses tulemust näidata, töödeldes vajalikud väljad rakenduse siseselt JavaScript-i funktsioonidega. Kuid kahjuks ei ole see käesoleva töö raames võimalik, nagu on hiljem täpsemalt selgitatud peatükis Andmed. Seega töös kasutatakse andmebaasi, ning selle andmebaasi juhtimissüsteemiks PostgreSQL-i, üks enamlevinumaid ning sellise relatsiooniliselt lihtsa andmebaasi jaoks väga sobiv. Valitud andmebaasi töötlemiseks kasutatakse programmeerimiskeeli Python-it, mis saab suurte koguste andmetega kiiresti ja tõhusalt hakkama ning JavaScript-i, mida saab kasutada rakenduse siseselt. Andmebaasi majutamiseks on praeguse seisuga väga palju valikuid, ka täiesti mitte tasulisi. Kuna rakendus ei ole ülemäära keeruka otstarbe ja nõudmistega, siis sobivad selleks enamus majutusteenused. Käesolevas töös kasutatakse selleks Supabase¹² platvormi.

Välja toodud veebiraamistikel on kõigil omad plussid ja miinused, kuid samas on need ka väga sarnased. Antud töös valiti veebiraamistikuks Vue.js. Seda valikut toetavad mitmed

¹² <https://supabase.com>

aspektid mis on Vue.js-ile omased, nagu näiteks väiksemaks projektiks suurem sobivus, kus projekti on lihtne üles seda ning arendada, kuid mis lubab ka edasiarenduste korral seda rakendus skaleerida märkimisväärselt. Reaktiivne olemus kajastab otsingutulemused tulemuste korral kasutajale koheselt ning seda ka lehti vahetades, iga leht ei pea uuesti algusest laadima. Autorile isiklikult meeldib ka Vue.js-i kogukond ja selle aktiivsus, mis esinenud probleemide korral aitab kiiresti leida vastuseid kas suurepärasest dokumentatsioonist, erinevatest foorumitest või Vue.js versioonihaldussüsteemi probleemide sektsioonist. Aastal 2021 valminud bakalaureusetöös võrdles Breedis põhjalikult samu veebiraamistikke ning jõudis sarnase järelduseni, kus kõige peale jäi Vue.js oma kasutajasõbralikkuse ja kogukonna pärast, lisaks sellele ka Vue.js-i olemuslik kesktee React-i ja Angular-i vahelise raamistikuna, mis kaasab endas positiivseid punkte mõlemast [26].

Lisaks Vue.js-i tugevustele on olemas ka selle alamraamistik Nuxt.js¹³, mis arendab selle võimekust ja arendamise mugavust veelgi rohkem. Nuxt.js on avatud lähtekoodiga raamistik, mis võimaldab kiiret, SEO (ingl *Search Engine Optimization*) sõbralikku ja skaleeritavat arendust serveri poolel renderdamise ning staatilise saidi genereerimisega [27]. Selle automatiseeritud marsruutimine, koodi tükeldamine ning eellaadimine hõlbustavad arendusprotsessi, millele lisaks TypeScript-i toetus, automaatne importimine ning rakenduse sisemine server teevad sellest võimsa alamraamistiku modernsete veebirakenduste loomiseks.

Veebipõhise rakenduse avalikustamiseks kasutatakse platvormi, kuhu veebirakendus laadida ning kust seda saab alati kättesaadavalt jooksutada. Selle töö raames kasutatakse selleks Netlify¹⁴ platvormi, mis ühildub väga hästi Nuxt.js-iga ning millel on tasuta majutamise võimalus kergemate rakenduste jaoks. Veebirakendus ise võib olla nii serveri pool renderdatud, kui ka staatilise saidina genereeritud. Antud töös valiti selleks SSR, sest andmete pärimisel ja nende töötlemisel tuleb kasuks serveri võimekus, mis kasutajatele kiiremini vastuse annab.

¹³ <https://nuxt.com>

¹⁴ <https://www.netlify.com>

5. Lahendus

Antud peatükis antakse ülevaade bakalaureusetöö käigus valminud lahendusest, mis on eestikeelne, avalik ning kasutajasõbralik. Samuti peatükis kirjeldatakse veebirakenduse valmimisprotsessi. Veebirakendus on ligipääsetav aadressilt <https://ravimite-koostoime.netlify.app/> ja selle koodibaas GitHub-is aadressilt https://github.com/vidrikt/ravimite_koostoime.

5.1 Andmed

Kui töö autor märkas enda ja lähedaste juures probleemi, et ravimite koostoimeid ei saa kiiresti ja kergelt kontrollida, tekkis idee luua taoline veebirakendus. Kuid enne teema paika panemist oli vaja leida andmed mida veebirakenduses kuvada. Eesti Ravimiameti lehel <https://www.ravimiregister.ee> saab hõlpsasti alla laadida erinevaid avalikke tabeleid ravimite, toimeainete, ATC koodide jms kohta. Kuid neil puuduvad ravimite koostoimete andmed eraldi andmekogudena, need on vaid käsitsi kirjutatud iga ravimi pakendi infolehele. Töö autor kontakteerus ka Ravimiameti andmehalduriga, kes kinnitas, et neil puuduvad selliselt kogutud andmed.

Eestis ainult tervishoiutöötajatele mõeldud ravimite koostoimete kliiniline otsusetugi kasutab sisemiselt Soome ettevõtte MedBase andmebaase INXBASE¹⁵ ja RISKBASE, varasema ühise nimetusega SFINX-PHARAO. Suheldes MedBase-iga otse, sai töö autor vastuseks, et nende Balti riikide turustajaks on Synbase ning ainult nende loal oleks andmebaase võimalik kasutada. Edasi suheldes Synbase tegevjuhiga, oli vastus eitav, sest see läheks vastuollu nende endi ning nende koostööpartnerite intellektuaalomandi õigustega.

Teise valikuna leidis töö autor USA riikliku meditsiinikogu RxNav API¹⁶. Need on avalikud andmed, küll päringute limiitidega, kuid siiski käesoleva töö tarvis sobivalt. Probleemideks aga osutusid asjaolud, et kõik vasted oleksid inglise keeles ning see andmebaas ei sisaldaks kõiki Euroopas ja Eestis kasutusel olevaid ravimeid. Arvestades nende väikeste puudujääkidega, alustas töö autor lahendusega. Kahjuks aga töö tegemise ajal, 2023. aasta novembris, teatati, et see API ühendus suletakse [28].

¹⁵ <https://www.medbase.fi/en/professionals/inxbase/>

¹⁶ <https://lhncbc.nlm.nih.gov/RxNav/APIs/>

Edasisel otsimisel ei leitud enam ühtegi avalikku ravimite koostoitete API ühendust, valikuks jäid vaid mõned andmebaasid. RxNav ravimite koostoitete API andmete üheks aluseks oli ettevõtte DrugBank¹⁷ andmebaas, mis on eraldiseisvalt siiani saadav ja akadeemilise loaga kasutatav. Kuid see on ääretult mahukas ja keerulise ülesehitusega andmebaas, millest ainult väike osa on ravimite koostoitete kohta. Samuti sisaldab see väheste sõnadega kirjeldust ning ilma kliinilise tähtsuse kategooriata, mis on antud rakenduse ülesehituse jaoks kriitiline. Läbides veel teisi erinevate puudujääkidega andmebaase, otsustas töö autor kasutada Norra riikliku meditsiinivahendite ameti NOMA andmebaasi¹⁸. Kuigi selle andmebaasi tulemused on norrakeelsed, mis lisab rakendusele kompleksust, on see-eest andmebaasil olemas kõik loodava veebirakenduse efektiivsemaks toimimiseks vajalikud väljad: kliinilise tähtsuse kategooria, koostoime tagajärje kirjeldus, olukorra kriteerium ning abinõu.

5.2 Andmete töötlemine

Järgmisena sammuna oli vaja need andmed veebirakendusel kasutamiseks sobivaks töödelda. See tähendas alla laetud andmebaasi tõmmise (ingl *database dump*) läbimist programmeerimiskeele Python abil, kasutades mooduleid *xml.etree.ElementTree* ja *xml.dom.minidom*. Iga ravimi koostoime kirje kohta andmebaasist salvestati uude XML (ingl *Extensible Markup Language*) faili väljad *Severity*, *SituationCriterion*, *ClinicalConsequence*, *Instructions* ning ravimite grupid *DrugGroup*, mille vahel see koostoime esineb. Väljastatud XML faili skeem (ingl *schema*) on nähtaval Joonis 1.

Seejärel kasutades programmeerimiskeelt JavaScript, loeti saadud XML fail PostgreSQL andmebaasi. Kasutati teeki *fs* failide lugemiseks, *csv-parser* CSV (ingl *Comma-Separated Values*) faili läbimiseks ning *xml2js* XML faili sisu JavaScript objektideks teisendamiseks. Saadud objektid läbiti ning loeti PostgreSQL andmebaasi *Prisma ORM*¹⁹ abil, mis on kaasaegne andmebaasi ORM (ingl *Object-Relational Mapping*) tööriist JavaScript-ile. Joonis 2 on kujutatud andmebaasi relatsiooniline mudel ning selle kolm tabelit *interactions*, *interaction_drug_groups* ja *drugs*, ehk koostoimed, koostoime ravimigrupid ning ravimid. Tabelil *interactions* on seos tabeliga *interacts_drug_groups*, mis hõlmab mõlemat selle

¹⁷ <https://www.drugbank.com>

¹⁸ Kasutatav aadressilt <https://www.legemiddelsok.no/sider/Interaksjoner.aspx>

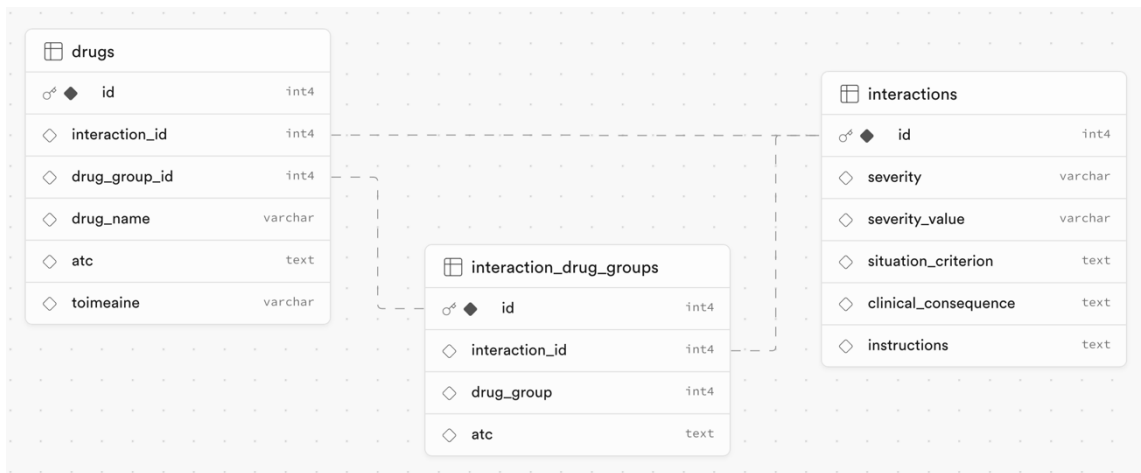
¹⁹ <https://www.prisma.io/orm>

koostoimega seotud ravimite gruppi, mille vahel koostoime esineb. Tabelil *interactins_drug_groups* on omakorda seos *drugs* tabeliga, kus on kõikide ravimite kohta käiv info. Lisatud on ka seos tabelite *drugs* ja *interactions* vahele, et oleks võimalik otse viidata ravimilt koostoimeni, milles see esineb.

Eelnevalt nimetatud CSV fail on Ravimiregistri lehelt alla laetud ATC koodide nimistu, kus iga ATC koodiga on seotud eestikeelne toimeaine nimetus. See tähendas, et olenemata võõrkeelsetest ravimite nimetustest algses andmebaasis, kirjutati uude rakenduse andmebaasi kõik ravimid eestikeelsete ravimite nimetustega, sobitades vastava ATC koodi kahe andmekogu vahel. Andmebaas oli arenduse faasis lokaalselt jooksev.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.kith.no/xmlstds/eresept/m30/2014-12-01">
  <xs:element name="DrugInteractions">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="Interaction">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" name="Severity">
                <xs:complexType>
                  <xs:attribute name="V" type="xs:unsignedByte"/>
                  <xs:attribute name="DN" type="xs:string"/>
                </xs:complexType>
              </xs:element>
              <xs:element minOccurs="0" name="SituationCriterion" type="xs:string"/>
              <xs:element name="ClinicalConsequence" type="xs:string"/>
              <xs:element minOccurs="0" name="Instructions" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element minOccurs="2" maxOccurs="2" name="DrugGroup">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" name="Drug">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="DrugName" type="xs:string"/>
                    <xs:element minOccurs="0" name="Atc">
                      <xs:complexType>
                        <xs:attribute name="V" type="xs:string"/>
                        <xs:attribute name="S" type="xs:string"/>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Joonis 1. Töödeldud andmete XML faili skeem.



Joonis 2. Andmebaasi relatsiooniline mudel.

Lisana esialgsest andmete töötlemisest, tuuakse siinkohal välja ka teised andmetega seotud protsessid, mis läbitakse iga otsinguga veebirakenduses. Kuna algsed andmed on norrakeelsed ning terve suure andmehulga tõlkimine oleks liiga protsessi rohke ning kallis, siis tõlkimise parimaks lahenduseks osutus reaalajas tõlge. Iga otsingu vastuse korral andmebaasist, päritakse tõlge väliselt teenuselt, milleks on Google Cloud Translate API. Kuid kuna see teenus pole lõplikult tasuta, siis päringute vähendamiseks salvestatakse juba tõlgitud otsingud rakenduse vahemällu. Teenusega suhtlemiseks kasutatakse JavaScript-i teeki *axios*.

Teise protsessina päritakse taaskord andmeid Ravimiregistrist. Lehelt alla laetud pakendite CSV failis on iga ravimi kohta selle pakendi infolehe alamkataloogi nimetus. Ehk lisades aadressile <https://ravimiregister.ee/Data/PIL/> saadud alamkataloogi nimetuse, näiteks „PIL_1671780.pdf“, tuleb vastuseks vastava ravimi pakendi infoleht PDF (ingl *Portable Document Format*) formaadis. Sellest dokumendi „Muud ravimid ja ...“ sektsioonist on võimalik leida lisainformatsiooni ravimite koostoimete kohta. Ehk õiges formaadis pakendi infolehe olemasolul, mis alati ei esine, näitab veebirakendus selle sektsiooni väljavõtet. Veebipõhine rakendus kaasab ka tervikliku pakendi infolehe lingi viiteks ning edasiste ohutusega seotud küsimuste lahendamiseks.

5.3 Veebirakendus

Lokaalsete töödeldud andmete olemasolul sai edasi liikuda veebirakenduse enda arendamisele. Algse projekti ja struktuuri ülesehituse jaoks kasutati Nuxt.js-i käsureainterpretaatorit *nuxi*, millega on võimalik genereerida projekti algne baas, koos vajalike kaustade, failide ning konfiguratsioonidega.

Pärast projekti seadistamist ja isikupärastamist oli esimeseks ülesandeks andmete ühendamine. Peale uurimist ja valikutega tutvumist, otsustati kasutada selleks teeki *Prisma ORM*. See teek võimaldab mugavalt suhelda andmebaasiga lisades sellele TypeScript-i tüüpimise toe ning lihtsa päringute ehitamise funktsionaalsuse. Kokku tehti seda teeki kasutades veebirakenduse valmimise jooksul kaks töötlejat (ingl *handler*) rakenduse serveri poolele:

- *drug-list.ts* - pärib andmebaasi *drugs* tabelist kõik eestikeelsed toimeained, mida kasutakse otsinguväljade automaatseks sõna lõpetuseks (ingl *autocomplete*);
- *interactions.ts* - otsib andmebaasist kahe või enama ette antud ravimite koostoimed, läbides kõik kolm andmebaasi tabelit.

Järgmiseks alustati veebirakenduse visuaalse poolega, ehk pandi paika stiilid – rohkesti kasutatavate stiilide muutujad ning komponendid. Selleks kasutati CSS-i (ingl *Cascading Style Sheets*) eelprotsessori skriptimiskeelt *SCSS*, mis laiendab CSS-i kasulikkust, pakkudes stiilide muutujaid ja komponente, mida saab kergesti korduv kasutada. Vastavad stiilid, nagu värvid, mõõdud, teksti vormid jms, saadi mugavalt eelnevalt koostatud Figma prototüübi pealt. Samuti koodi kordamise vähendamiseks pakub Nuxt.js paigutusfailide võimalust, mis tähendab üldise kujunduse defineerimist – pealkirjad, jalused ja navigatsioon, ehk mitmetel lehtedel korduvad elemendid. Antud tööd tehti üks paigutusfail, kuvatud Joonis 3, mida kasutati kõikidel lehtedel.

```
1 <template>
2   <div class="layout-default">
3     <CommonHeader :without-logo="withoutLogo" />
4     <slot />
5     <CommonFooter :without-disclaimer="withoutDisclaimer" />
6   </div>
7 </template>
```

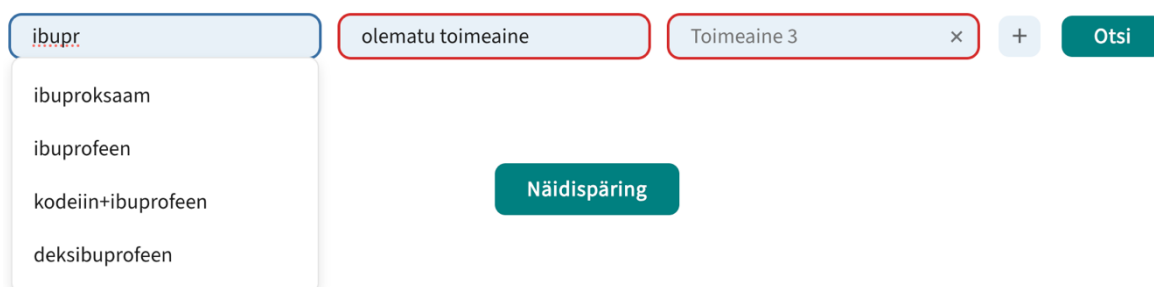
Joonis 3. Paigutusfail korduvatele elementidele.

5.3.1 Veebirakenduse vaated

Alustuseks valmis veebirakenduse avalehe vaade, kus esineb esimesena otsingu sektsioon. Selles on algselt kaks tekstivälja, mida on võimalik juurde lisada, vajutades nupul "+", kuni kahe välja ulatuses, ehk kokku neli välja. Neid lisatud väljasid on ka võimalik eemaldada, vajutades sümbolil "x". Kõik tekstiväljad peavad olema täidetud, ei tohi korduda ning

sisestatu peab vastama õigele toimeainele. Ebakorrektsuse korral annavad tekstiväljad veast märku ümbritseva punase servaga. Valideerimisel kontrollitakse sisestatud teksti andmebaasi *drugs* tabeli kirjete vastu. Sama tabeli toimeainete nimekiri kuvatakse ka tekstivälja alla automaatseks sõna lõpetuseks, kui kasutaja on sisestanud vähemalt ühe sümboli. Need toimeained on korrektsuse eesmärgil kuvatud selliselt, nagu need on Ravimiregistri enda andmetes – mõningad toimeained on suurte algustähtedega ning mõned tervenisti kapiteelkirjas. Vajutades nupul “Otsi” viiakse kasutajat täidetud otsingu terminitega vahelehele „Otsing“, kus teostatakse automaatselt päring. Joonis 4 on kujutatud otsingu sektsiooni erinevate olekutega ning automaatse sõna lõpetuse aken.

Avalehe disainis jälgiti enamusest prototüüpi, kuid lisati otsingu sektsiooni alla näidispäringu nupp, mis viib „Otsingu“ vahelehele näidisotsingu terminitega.




The image shows a search interface. On the left, there is a text input field containing 'ibupr'. Below this field is a dropdown menu with four suggestions: 'ibuproksaam', 'ibuprofeen', 'kodeiin+ibuprofeen', and 'deksibuprofeen'. To the right of the input field are two more input fields: one labeled 'olematu toimeaine' and another labeled 'Toimeaine 3' with a clear button (x). To the right of these fields is a plus sign (+) and a green button labeled 'Otsi'. Below the input fields is a green button labeled 'Näidispäring'.

Joonis 4. Otsingu sektsioon.

Tulemuste kuvamiseks valmis vaheleht „Otsing“. Vaatel on esimesena sama otsingu sektsioon. Otsingu toimumisel kuvatakse kasutajale animeeritud liikuvad nooled, et kinnitada kasutajale otsingu toimumist. See on praegu rakenduse juures vajalik, sest tasuta majutatud veebirakendus ja andmebaas ei anna just kõige kiiremini tulemusi. Kui otsingu tulemuste korral kuvatakse kasutajale kas „Koostoimeid ei leitud.“ või „Leitud koostoimed:“ mille järele lisatakse tulemused. Tulemused kuvatakse üksteise alla, kus iga sektsioon näitab kahe toimeaine vahelist koostoimet, nähtav Joonis 5. Üks koostoime sisaldab kliinilise tähtsuse kategooriat, koostoime tagajärje kirjeldust, olukorra kriteeriumit ning abinõud, kus viimased kaks ei ole kohustuslikud, kõikidel koostoimetel neid ei esine. Viimasena võib vastavalt koostoimele olla Ravimiregistri pakendi infolehel tekst ja link tervele infolehele. Ravimiregistri igal toimeainel pakendite infolehte ei esine. Kui mõnedel infolehtedel puudub koostoime sektsioon või on seda sealt tekstina raske välja võtta, siis tulemuses on kuvatud ainult link infolehele. Nagu igal veebirakenduse vaatel, on ka siin alati nähtav hoiatus veebirakenduse mittetäieliku meditsiinilise olemuse kohta.

LEITUD KOOSTOIMED:

krisotiniib  flukonasool	
Tuleks vältida	
Kliiniline tagajärg	Krisotiniibi suurenenud kontsentratsioon.
Olukorra kriteerium	Kehtib flukonasooli annustamise kohta sagedamini kui üks kord nädalas, mitte flukonasooli ühekordse kasutamise kohta.
Abinõu	<p>Annuse kohandamine: Krisotiniibi annusevajadus on hinnanguliselt ligikaudu 70–80% madalam flukonasooli kombineeritud ravi ajal, kuid koostoime määr on väga erinev. Seetõttu on oluline hoolikas kliiniline jälgimine. Krisotiniibi tootja kirjutab Xalkori pakendi infolehes, et samaaegset kasutamist tuleks vältida.</p> <p>Ravimi alternatiivid: tuleb hinnata individuaalselt.</p>
Koostoime Ravimiregistri pakendi infolehel	<p>Teatage oma arstile või apteekrile, kui te võtate või olete hiljuti võtnud või kavatsete võtta mis tahes muid ravimeid.</p> <p>Informeerige kohe oma arsti, kui te kasutate astemisooli, terfenadiini (antihistamiinsed ravimid allergia raviks) või tsisapriidi (kasutatakse seedeprobleemide raviks) või pimosiidi (kasutatakse vaimsete häirete raviks) või kinidiini (kasutatakse südame rütmihäirete raviks) või erütromütsiini (antibiootikum infektsioonide raviks), kuna neid ei tohi võtta koos ravimiga Diflucan mg/ml (vt lõik „Ärge kasutage Diflucan mg/ml“).</p> <p>Link pakendi infolehele</p>

ⓘ See leht on teoreetiline ning ükski saadud tulemus ei ole lõplik meditsiiniline vastus. Oluliste küsimuste ja probleemide korral pöörduge meditsiinilise haridusega spetsialisti juurde!

Joonis 5. Vahelehe „Otsing“ tulemuse osa.

Viimase vaadena veebirakenduses tehti „Abi“ vaheleht. See vaade aitab kasutajat küsimuste korral ning annab informatiivset ülevaadet veebirakendusest. Kuvatud on veebirakenduse loomisega seotud taust, ehk et see valmis autori bakalaureusetöö raames ning mis andmeid on rakenduses kasutatud. Lisatud on ka link rakenduse koodibaasile. Vaheleht selgitab kasutajale ravimite koostoime olemusest ning selle ohtlikkusest. Välja on ka toodud rakenduse kasutamise õpetus ning mida tulemused võivad sisaldada. Vahelehe „Abi“ on nähtav vahelehe „Abi“ sisuline osa.

Kui kõik vaated olid valmis, tehti veebirakendus mobiilisõbralikuks. Rakenduse arenduse käigus mõeldi sellele juba osaliselt ette, seega mobiilivaates säilitas veebirakendus enamuse stiilide ja vormi korrektsuse, nõudes väheseid parandusi ja lisandusi. Kuid siiski oli vaja lisada eraldi mobiilimenüü, mis asendab tavalist menüüd, peites avanemise päises oleva nupu taha.

5.3.2 Veebirakenduse avalikustamine

Peale rakenduse valmimist, oli see vaja avalikustada. Nuxt.js-i dokumentatsioonis pakutakse koheselt variante, milliseid platvorme võiks kasutada ning töö autor valis selleks Netlify platvormi. Ühendades soovitud versioonihaldus süsteemi ning selle sisese projekti, jooksub platvorm ise avalikustamise protsessi peale igat koodi üles laadimist. Ainsaks probleemiks osutus peale avalikustamist JavaScript-i teek *pdf-parser*, mida Ravimiregistri pakendi infolehtede töötlemiseks kasutati. Peale otsimist leidis, et sarnane probleem esineb ka teistele teegi kasutajatel, kus see teek teadmata põhjustel proovis jooksubada sisemisi testfunktsioone, mis aga ebaõnnestusid ning koodi ehitamise katkestasid. Selle lahenduseks pidi rakendusele lisama uue teegi *patch-package*, millega sai muuta teekide sisemist koodi. Nimelt sunniti vigast teeki testfunktsioone mitte jooksumata. Kuna manuaalsed teekide koodi mõjutused ei ole püsivad, siis *patch-package* aitab neid muudatusi sundida peale igat teekide installeerimist.

Kuna veebirakendus hakkas jooksuma serveris, siis seda oli vaja ka teha andmebaasi jaoks, ehk majutusteenuseks kasutati avatud lähtekoodiga platvormi Supabase. Ühenduse jaoks kasutati taaskord *Prisma ORM* teeki, kus tehti ümber varasemalt lokaalsesse andmebaasi laadinud skripti loogika, et andmed jõuaksid eemal asuvasse majutatud andmebaasi. Lisaks on Supabase platvormil andmebaaside majutamise kõrvalt võimalus tavapäraseid üksik faile või kaustu hoida. See tuli vägagi kasuks, sest nii sai sinna laadida Ravimiregistrist võetud pakendite loendi. Seega sai rakendus pakendi infolehti küsida nüüdsest otse sisuedastusvõrgult, ehk CDN-ilt (ingl *Content Delivery Network*), ilma seda rakenduse oma failides hoiustades ja ehituse mahtu suurendades.

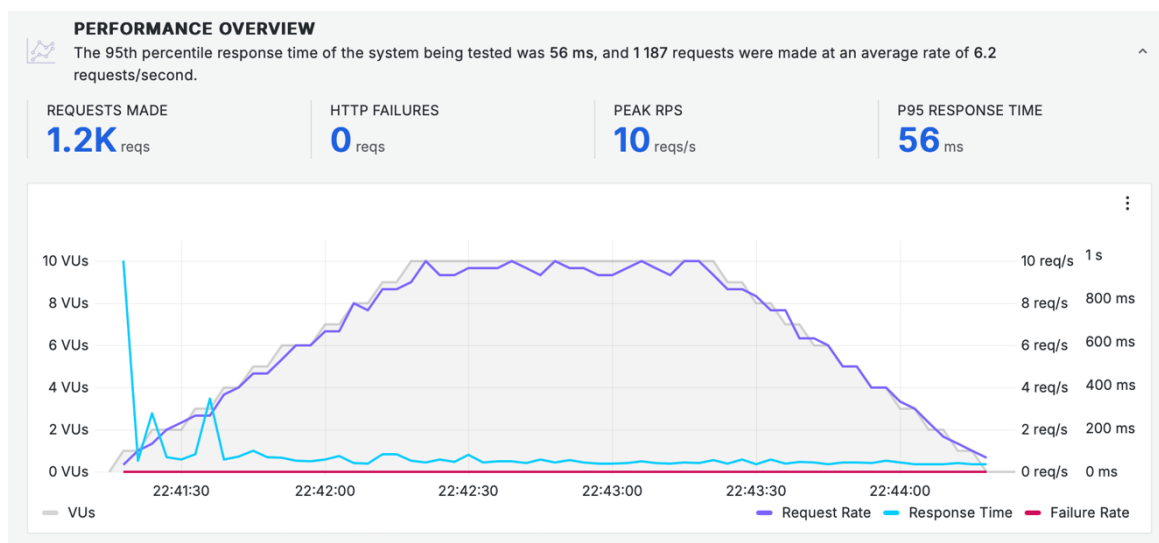
6. Lahenduse analüüs ning kasutajasõbralikkuse testimine

Selles peatükis analüüsitakse valminud lahendust ning kontrollitakse valminud veebirakenduse vastavust nõuetele. Samuti kirjeldatakse kasutajasõbralikkuse testimist ning analüüsitakse selle tulemusi.

6.1 Vastavus nõuetele

Veebirakenduse kõik funktsionaalsed ja mittefunktsionaalsed nõuded on täidetud. Järgnevalt on eraldi välja toodud kolme mittefunktsionaalse nõude testi tulemused.

REQ 24 – „Veebirakendus peab suutma hallata suurenenud kasutajate koormust, ehk vähemalt 10 samaaegset kasutajat, ning suuremat andmemahutu ilma olulise jõudluse halvenemiseta ning hangumiseta.“ Selle nõude kontrollimiseks kasutati koormuse testimise platvormi k6²⁰ ning selle ühilduvust Grafana Cloud-iga²¹, et näidata tulemusi visuaalselt. Tulemused näitasid, et loodud veebirakendus sai hakkama kümne samaaegse kasutajaga ilma päringute aja suure tõusmiseta, nähtav Joonis 6.

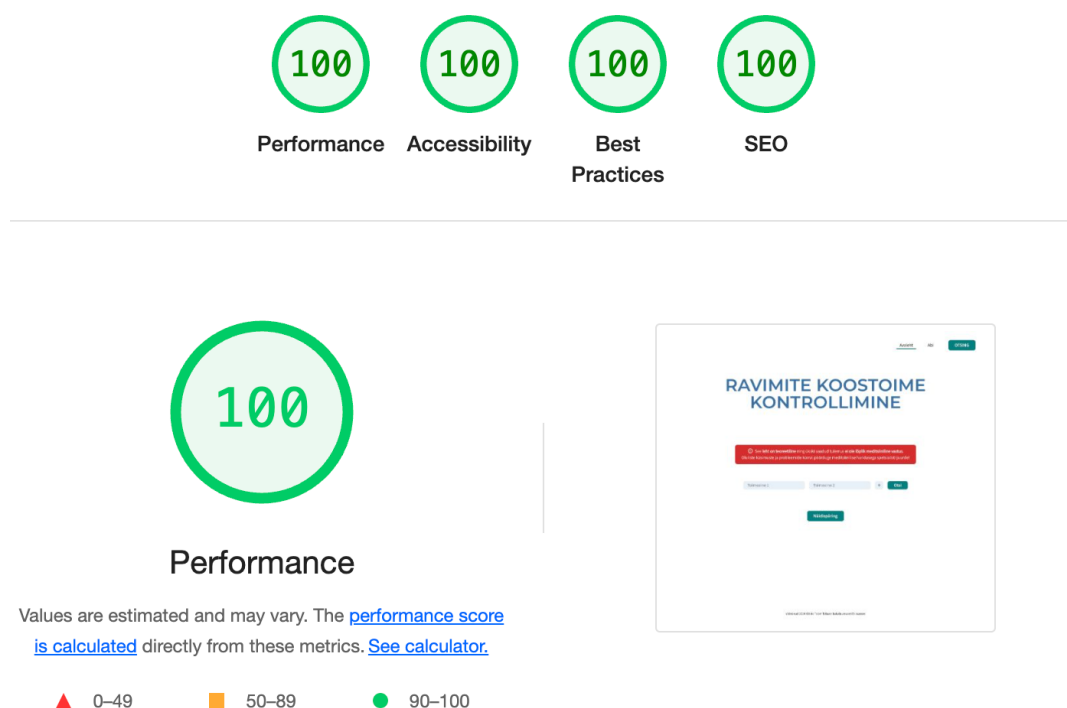


Joonis 6. Veebirakenduse koormuse test platvormil k6.

²⁰ <https://k6.io>

²¹ <https://grafana.com/products/cloud/k6/?src=k6io>

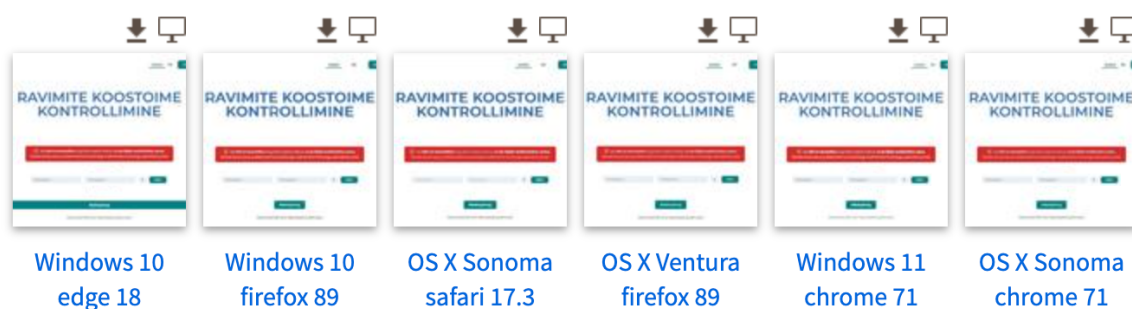
REQ 26 – „Veebirakenduse Google Lighthouse skoor peab olema üle 90-e kõikides kategooriates. Seda mõlema ekraani suurusega, *desktop* ning *mobile*.“ Selle nõude kontrollimiseks kasutati Google Chrome DevTools-i integreeritud Lighthouse tööriista. Loodud veebirakendus saavutas maksimaalse tulemuse kõikides kategooriates, seda nii mõlemal ekraani suurusel kui ka igas vaates, nähtav Joonis 7.



Joonis 7. Veebirakenduse Google Lighthouse skoor.

REQ 28 – „Veebirakendus peab ühilduma veebibrauseritega Google Chrome, Mozilla Firefox, Safari ja Microsoft Edge ning nende hiljutisemate versioonidega.“ Selle nõude kontrollimiseks kasutati BrowserStack Screenshot²² testimise platvormi. Tööriist võimaldab valida erinevate seadmete, veebibrauserite ja nende versioonide kombinatsioone, mille põhjal genereeritakse veebilehest kuvatõmmised. Neid visuaalselt võrreldes saab aimu, kuidas veebirakendus erinevatele kasutajatele välja paistab. Loodud veebirakendus tundub testi tulemuste põhjal töötavat kõikides nõuetes välja toodud veebibrauserites, nähtav Joonis 8.

²² <https://www.browserstack.com/screenshots>



Joonis 8. BrowserStack Screenshot testi tulemus [29].

6.2 Kasutajasõbralikkuse testimise

Süsteemi kasutatavuse mõõdik, ehk SUS (ingl *System Usability Scale*), mille arendas välja John Brooke [30], pakub võimalust erinevate süsteemide üldise kasutatavuse hindamiseks. Brooke-i 1996. aastal avaldatud artikkel kirjeldab SUS-i tagamaid ja selle rakendamist professionaalsete süsteemide kasutatavuse hindamiseks. Kasutatavus, nagu Brooke rõhutab, pole absoluutne omadus, vaid sõltub süsteemi kasutamise kontekstist.

SUS on kümne väitega küsimustik [30], mis on mõeldud kasutajate subjektiivsete kasutatavuse hinnangute kiireks ja usaldusväärseks kogumiseks. Selle väited hõlmavad erinevaid kasutatavuse aspekte, nagu abi vajadus, õppimine ning keerukus. Skaala põhineb Likerti skaala²³ metodoloogial, kus vastajad hindavad iga väidet nõusolekuga või vastuseisuga 5-palli süsteemis. Süsteemi kasutatavuse mõõdik on tänapäevani laialdaselt kasutusel tänu oma lihtsusele, usaldusväärsusele ja kohanemisvõimele, muutes selle populaarseks valikuks süsteemi kasutatavuse hindamiseks erinevates tööstusharudes ja uurimisvaldkondades [31].

SUS-i ülesehitus ja kasutamine on lihtne: vastajad hindavad iga väite punkti kohe pärast süsteemi kasutamist. Üksikute punktide hinnangud pole iseenesest tähendusrikkad, vaid need panustavad üldise kasutatavuse mõõdiku moodustamisse. SUS-i tulemus, mille vahemik 0 kuni 100, esindab seda kokkuvõtlikku mõõdikut ja pakub kasutatavuse kvantitatiivset hinnangut [30].

²³ https://en.wikipedia.org/wiki/Likert_scale

Käesoleva töö kasutajasõbralikkuse testimiseks koostas töö autor küsimustiku Google Forms-is²⁴ ja tõlkis süsteemi kasutatavuse mõõdiku väited eesti keelde:

1. Ma arvan, et mulle meeldiks seda süsteemi sageli kasutada.
2. Leidsin, et süsteem on tarbetult keeruline.
3. Mulle tundus, et süsteemi oli lihtne kasutada.
4. Arvan, et mul oleks vaja tehnilise inimese tuge, et seda süsteemi kasutada.
5. Leidsin, et erinevad funktsioonid selles süsteemis olid hästi integreeritud.
6. Mulle tundus, et selles süsteemis oli liiga palju ebajärjekindlust.
7. Ma kujutaksin ette, et enamik inimesi õpiks selle süsteemi kasutamise väga kiiresti.
8. Leidsin, et süsteemi kasutamine oli väga ebamugav.
9. Tundsin end süsteemi kasutades väga enesekindlalt.
10. Mul oli vaja palju õppida, enne kui seda süsteemi kasutada sain.

6.2.1 Kasutajasõbralikkuse testimise tulemused

Küsitlus saadeti 12 inimesele. Töö autor palus vastajatel kõigepealt tutvuda veebirakendusega ning seejärel täita küsimustiku. Vastamine oli anonüümne ning toimus testijate oma valitud seadmetes.

Süsteemi kasutatavuse mõõdiku, tulemuseks tuli 87.5²⁵ punkti 100-st. Saavutatud tulemus asub vahemikus 80.3 kuni 100, mis tähendab, et valminud veebirakenduse kasutatavus on suurepärane, vastab hindele A [32]. Lisaks on saavutatud tulemus märkimisväärselt üle keskmise, milleks on 70 punkti [33]. Seega tulemused näitavad, et vastajate hinnangul on rakendus kõrge kasutatavusega - rakendus on funktsionaalne ja hea kasutajakogemusega.

²⁴ <https://forms.gle/HZjD3cpULHtnFJQH6>

²⁵ https://docs.google.com/spreadsheets/d/1Ioz9i0LIonvte9vZ2BHAYpPpa2d2py_dteE5P6ZQQsg

6.2.2 Vastajate kommentaarid

Kasutajasõbralikkuse testimise küsimustikus oli ka lisa väli kommentaaridele, kuhu sai valikuliselt jätta tagasisidet. Vastajate antud tagasiside:

- „Süsteem ise on loogiline ja seda on lihtne kasutada. Võiks veel eraldi lehena olla loend levinumatest toimeainetest ja ravimitest, kus need leiduvad.“;
- „Idealis sooviks otsida ravimeid ka tootenimega mitte ainult toimeaine järgi.“;
- „Otsingukast võiks anda ka ligilähedasi tulemusi teksti trükkides. Juhul kui ei tea täpset toimeainet, aga enamvähem mäletad, siis otsing aitab lähedase vaste leida ka siis kui teed kirjavea.“;
- „Näidispäringu nupp võiks rohkem erineda "Otsi" nupust UX'i seisukohalt. Ehk muuta paigutust või lisada näidispäringu nupule juurde seletav tekst. Kuidagi intuiivselt tahtsin rohkem seda näidipäringu nuppu vajutada, sest see on otsi nupust suurem ja asub väga kesksel positsioonil.“.

Otsene tagasiside kasutajatelt on rakenduse valmistamisel väga olulisel kohal ning seetõttu on ka osad nendest kasulikest kommentaaridest välja toodud järgmises peatükis, edasiarengu võimalustena.

7. Edasised arenguvõimalused

Bakalaureusetöös loodava veebirakenduse eesmärk oli pakkuda eestikeelset ning kättesaadavat ravimite koostoime kontrollimise võimalust. Loodud veebirakendus täitis selle eesmärgi, kuid on mõned edasiarengu suunad, mis tõstaksid rakenduse kasulikkust veelgi rohkem.

Veebirakendusele saaks lisada pakendi nimetuse järgi otsimise võimaluse. Kuna kasutajad ei pruugi teada kindla ravimi toimeaine nimetust või veel vähem ATC koodi, siis praeguse rakenduse kasutamiseks peavad nad ennemalt toimeaine nimetuse välja selgitama. Ravimiregistri enda otsing võimaldab sellist otsingut ning sealt saadud andmete põhjal oleks ka võimalik see töös loodud veebirakendusele lisada.

Veebipõhisel rakendusel võiks olla võimalus otsida vaid ühe toimeaine järgi, mille korral näidatakse tulemustes kõiki koostoimeid, mis selle ravimiga esinevad. Niimoodi saaks kasutaja ülevaate kõikidest teistest toimeainetest, millel esineb koostoime otsitud ravimiga.

Veebirakendusele saaks ka lisada ravimite ja toidu vahel esinevate koostoimete kontrolli. See tõstaks rakenduse olulisust, sest sellist tüüpi kontrolle on alternatiividena veelgi vähem, eriti arvestades avaliku ligipääsu kriteeriumit.

Rakenduse otsingu tagastuse kiirus võiks olla kiirem, mis teeks selle veebirakenduse kasutamise mugavamaks ning atraktiivsemaks. Kindlasti on võimalik otsingufunktsioone optimeerida rakenduse koodi siseselt, kuid hetkel on suurimaks probleemiks tasuta platvormide kasutamine. Nimelt veebirakenduse majutusteenus ning eriliselt veel andmebaasi majutus saavad olla potentsiaalselt kordades kiiremate päringute lahenduseks, kuid selline infrastruktuur on tasuline.

Kokkuvõte

Käesoleva bakalaureusetöö raames valmis veebirakendus ravimite koostoimete kontrollimiseks. Veebipõhine rakendus on ainsa avaliku eestikeelse alternatiivina teistele sarnastele lahendustele. Veebirakendus on ligipääsetav aadressilt <https://ravimite-koostoime.netlify.app/> ja selle koodibaas GitHub-is aadressilt https://github.com/vidrikt/ravimite_koostoime.

Bakalaureusetöö koosneb ravimite koostoime ja selle ohu tutvustamisest ning olemasolevate lahenduste välja toomisest, selgitades töö aktuaalsust. Töös pandi paika veebirakenduse funktsionaalsed ja mittefunktsionaalsed nõuded, lisades interaktiivse prototüübi. Toodi välja veebirakenduse loomiseks kasutatavad tehnoloogiad ning nende võrdlus alternatiividega. Kirjeldati veebirakenduse loomise protsessi ning sellele eelnevat eeltööd. Lõpus toodi välja veebirakenduse lahenduse analüüs, testimine ja tulemused ning edasiarengu võimalused.

Veebirakenduse ravimite koostoimete andmete saamiseks kasutati Norra riikliku andmebaasi NOMA. Andmete töötlemiseks olid kasutusel nii Python kui ka JavaScript programmeerimiskeeled, lugedes andmed PostgreSQL andmebaasi. Veebirakendus ehitati veebiraamistiku Vue.js ning selle alamraamistiku Nuxt.js peale. Lahendus vastab püstitatud nõuetele ning on testgrupi tulemuste põhjal väga funktsionaalne ning kasutajasõbralik.

Viidatud kirjandus

1. Vallaste H. e-Teatmik: IT ja sidetehnika seletav sõnaraamat. <http://www.vallaste.ee/>
2. Wiegers KE, Beatty J. Software requirements. Third edition. Redmond, Washington: Microsoft Press. 2013.
3. AKIT - Andmekaitse ja infoturbe leksikon. <https://akit.cyber.ee/>
4. Ravimiamet. Kõrvaltoimed. <https://www.ravimiamet.ee/ravimid-ja-ohutus/ravimiohutuskorvaltoimed> (22.04.2024)
5. Clinical Info. Drug-Drug Interaction. <https://clinicalinfo.hiv.gov/en/glossary/drug-drug-interaction> (22.04.2024)
6. Brown E. Web development with Node and Express. First edition. Sebastopol, CA: O'Reilly. 2014.
7. Cascorbi I. Drug Interactions — Principles, Examples and Clinical Consequences. *Deutsches Ärzteblatt International*, 2012, lk 546–556. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3444856/>
8. Tervisekassa. 2024. Suur osa Eesti elanikest tarbib mõnda retseptiravimit. <https://www.tervisekassa.ee/uudised/suur-osa-eesti-elanikest-tarbib-monda-retseptiravimit> (08.04.2024)
9. Varghese D, Ishida C, Haseer Koya H. Polypharmacy. *StatPearls*. Treasure Island, FL: StatPearls Publishing, 2024. <http://www.ncbi.nlm.nih.gov/books/NBK532953/>
10. Angamo MT, Chalmers L, Curtain CM, Bereznicki LRE. Adverse-Drug-Reaction-Related Hospitalisations in Developed and Developing Countries: A Review of Prevalence and Contributing Factors. *Drug Safety*, 2016, lk 847–857. <https://doi.org/10.1007/s40264-016-0444-7>
11. Raudsepp D, Lass J, Saar M, Kuld L, Irs A. Ravimite kõrval- või koostoimed erakorralise meditsiini osakonda pöördumise põhjusena – prospektiivne vaatlusuuring. *Eesti Arst*, 2011, nr 90, lk 122–128. <https://www.etis.ee:2443/Portal/Publications/Display/5d6f08e4-35a8-4812-a43c-6fc9d234fcbd>
12. Aronson JK. Communicating information about drug interactions. *British Journal of Clinical Pharmacology*, 2007, lk 637–639. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2000589/>
13. FEST Implementation guidelines. *The Norwegian Medical Products Agency*. 2024. https://www.dmp.no/globalassets/documents/om-oss/distribusjon-av-lege-middeldata/fest/hvordan-bruke-fest/202404_implementasjon-guide-fest-v3.5.pdf (22.04.2024)

14. Tervisekassa. Kaljuste D, Kuslapuu M. Eelvisiit Projekti mõju-uuring. *Tallinna Tehnikaülikool*, 2023. <https://www.tervisekassa.ee/media/588/download?inline> (14.04.2024)
15. Tamro Baltics. Apteekri virtuaalnõustamisel küsitakse kõige enam ravimite koostoime kohta. <https://www.tamrobaltics.com/uudised-en/apteekri-virtuaalnustamisel-ksitakse-kige-enam-ravimite-koostoime-kohta> (11.04.2024)
16. Tervisekassa. 2023. Ravimite koostoime otsusetugi. <https://www.tervisekassa.ee/partnerile/raviasutusele/kliinilised-otsusetoed/ravimite-koostoime-otsusetugi> (04.01.2024)
17. Tepandi J. Tarkvara protsessid ja kvaliteet. 2024. <https://tepandi.ee/tns-loeng.pdf> (14.02.2024)
18. Koníček I, Světlík J. Some principles for the design of successful and appealing web-sites. *European Journal of Media, Art and Photography*, 2020, lk 106–116. <https://ejmap.sk/some-principles-for-the-design-%e2%80%a8of-successful-and-appealing-websites/>
19. OSS Insight. Javascript Framework - Ranking. <https://ossinsight.io/collections/javascript-framework/> (01.05.2024)
20. Joshi M. Angular vs React vs Vue: Core Differences. 2023. <https://browserstack.wptengine.com/guide/angular-vs-react-vs-vue/> (01.05.2024)
21. Diniz-Junior RNV, Figueiredo CCL, De S.Russo G, Bahiense-Junior MRG, Arbex MVL, Dos Santos LM, et al. Evaluating the performance of web rendering technologies based on JavaScript: Angular, React, and Vue. *2022 XLVIII Latin American Computer Conference (CLEI)*, 2022, lk 1–9. <https://ieeexplore.ieee.org/document/9959901>
22. Kaur G, Tiwari RG. Comparison and Analysis of Popular Frontend Frameworks and Libraries: An Evaluation of Parameters for Frontend Web Development. *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2023, lk 1067–1073. <https://ieeexplore.ieee.org/document/10192987>
23. Vyas R. Comparative Analysis on Front-End Frameworks for Web Applications. *International Journal for Research in Applied Science and Engineering Technology*, 2022, lk 298–307. <https://doi.org/10.22214/ijraset.2022.45260>
24. Osmani A, Miller J. Rendering on the Web. *web.dev*. <https://web.dev/articles/rendering-on-the-web> (05.05.2024)
25. Nuxt. Deployment · Get Started with Nuxt. <https://nuxt.com/docs/getting-started/deployment> (05.05.2024)
26. Breedis R. JavaScripti kasutajaliidese raamistike võrdlus. TÜ arvutiteaduse instituudi bakalaureusetöö. 2021. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=72077
27. Nuxt.js. 2024. <https://github.com/nuxt/nuxt> (06.05.2024)

28. DrugBank Blog. 2023. NIH Discontinues their Drug Interaction API. <https://blog.drugbank.com/nih-discontinues-their-drug-interaction-api/> (19.04.2024)
29. Browserstack. Cross Browser Screenshots. <https://www.browserstack.com/screenshots/c35e5f3c099791f4eb6ae32e6bd54bf731a07228> (11.05.2024)
30. Brooke J. SUS: A quick and dirty usability scale. *Usability Eval. Ind.*, 1995.
31. Brooke J. SUS: a retrospective. *Journal of Usability Studies*, 2013, 1k 29–40. <https://uxpajournal.org/sus-a-retrospective/>
32. Sauro J. Measuring Usability with the System Usability Scale (SUS). <https://measuringu.com/sus/> (13.05.2024)
33. Bangor A, Kortum. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *JUX - The Journal of User Experience*, 2009, 1k 114–123. <https://uxpajournal.org/determining-what-individual-sus-scores-mean-adding-an-adjective-rating-scale/>

Lisad

I. Vahelehe „Abi“ sisu

SIIT LEIAD TEKKINUD KÜSIMUSTELE VASTUSEID

Lisaks informaativseid ja abistavaid viiteid välistele lehtedele

See veebirakendus on valminud 2024 bakalauresetöö raames.

Autor Vidrik Toom Tabas.

Juhendaja Lidia Feklistova.

Rakendus ning otsingutulemused on terveniisti eestikeelsed.

Andmed pärinevad Norra riikliku meditsiinivahendite ameti [NOMA](#) (Norwegian Medical Products Agency) andmebaasist. Kasutatud andmed on algselt norrakeelsed, seetõttu võib esineda mõningaid tõlkevigu.

Ravimiameti [Ravimiregistrist](#) on saadud eestikeelsed toimeaine nimetused ning pakendite infolehed.

[Koodibaas GitHub'is](#)

RAVIMITE KOOSTOIMEST

Ükski ravim pole lõplikult ohutu. Igal toimeainel on organismile mõju ning seetõttu on väga oluline ravimite õige kasutamine. Ravimite koostoime on erinevate toimeainete omavaheline vastastikmõju. Mitmete erinevate ravimite samaaegsel kasutamisel suureneb tõenäosus koos- ja kõrvaltoimete esinemisele. Need koostoimed võivad põhjustada mitmesuguseid soovimatuid tagajärgi, sealhulgas ühe või mitme ravimi terapeutiliste mõjude võimendumist või nõrgenemist, kõrvaltoimete suurenenud riski või ravimite toime muutust organismis. Ravimite tarvitajad peavad olema valvsad potentsiaalsete koostoimete tuvastamisel ja ettevaatusabinõude rakendamisel, et saavutada soovitud tulemused ravimite võtmisel ning tagada ohutus.

[Ravimiameti artikkel kõrvaltoimete kohta](#)

RAKENDUS

Ravimite koostoime kontrollimiseks peate sisestama otsingusse vähemalt kaks toimeaine nimetust või ATC koodi ning vajutada "Otsi".

Leitud ravimite koostoimed kuvatakse eraldi sektsioonidena, kus iga sektsioon sisaldab kahe ravimi koostoimet.

Tulemus annab esimesena hinnangu koostoime kliinilisele tähtsusele kategooriaga, mis on üks järgnevatest:

Tuleks vältida

Kõrge kliinilise tähtsusega.
Suur koostoime risk. Arst peaks olukorda regulaarselt jälgima, vajadusel kaaluda alternatiivset ravimit.

Tuleks rakendada ettevaatusabinõusid

Mõõduka kliinilise tähtsusega.
Võimalusel hoiduda kombinatsioonest, kasutades neid ainult erijuhtudel. Arsti jälgimine on tungivalt soovitatav.

Kliiniliselt ohutu koostoime

Minimaalse kliinilise tähtsusega.
Koostoime on ebatõenäoline, väike või ebaoluline.

Iga ravimi koostoime sektsioon sisaldab välja "Kliiniline tagajärg" ning võimalikke väljasid "Olukorra kriteerium" ja "Abinõu".

Lisaks on välja toodud info Ravimiregistri pakendi infolehelt ravimite koostoime kohta, kui see eksisteerib. Edasise huvi ja ohutuse tarvis on lisatud link tervele pakendi infolehele.

II. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Vidrik Toom Tabas**,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

Eestikeelne ravimite koostoime kontrollimise veebirakendus,

mille juhendaja on **Lidia Feklistova**,

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Vidrik Toom Tabas

15.05.2024