

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Kärt Allik 193978IADB

Veebirakendus kursuse Kõrgem matemaatika toetamiseks

Bakalaureusetöö

Juhendaja: Kristiina Hakk
PhD

Tallinn 2022

Autorideklaratsioon

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

Autor: Kärt Allik

16.05.2022

Annotatsioon

Kursus Kõrgem matemaatika on kohustuslik õppeaine kahele Tallinna Tehnikaülikooli IT Kolledži õppekavale: IT süsteemide arendus ja IT süsteemide administreerimine. See hõlmab endas kolme erinevat matemaatika valdkonda. Hetkel on kursuse materjalid kokku koondatud Moodle keskkonda, mis ei võimalda interaktiivselt ja piisavalt paindlikul viisil koos abistava materjalidega harjutusülesandeid lahendada.

Käesoleva bakalaureuse töö eesmärk on luua veebikeskkond, mis võimaldab tudengitel harjutada kursusega seonduvad ülesandeid koos teoreetilise toega, et materjali paremini omandada.

Töö koosneb teoreetilisest ja praktilisest osast. Teoreetilises osas analüüsitakse kursuse ülesehitust, piiratakse lõputöö skoop ja uuritakse interaktiivse õppekeskkonna loomise meetodeid. Lisaks võrreldakse erinevaid tehnoloogiaid lahenduse teostuseks.

Töö praktiline osa on vastava veebirakenduse arendamine. Tulemusena valmis veebirakendus, kus on võimalik lahendada lineaaralgebra ülesandeid. Tudengite seas läbiviidud küsitluses selgus, et rakendust on mugav kasutada ning pakub reaalselt väärtust õppimisprotsessis.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 45 leheküljel, 7 peatükki, 19 joonist, 2 tabelit.

Abstract

Web Application for Supporting the Subject Higher Mathematics

At Tallinn University of Technology, Higher Mathematics is a required course for two programmes: IT Systems Development and IT Systems Administration. It covers three mathematical topics: mathematical analysis, statistics and probability theory and linear algebra. All course materials are currently available at Moodle learning environment. Moodle provides a well-developed way to provide theoretical materials but lacks the ability to allow students to solve practical exercises within the environment. Existing solutions can be misleading, do not provide enough exercises or generally lack theoretical support, which is relevant from a learning aspect.

The aim of this thesis is to create a web-based environment which allows students to practice and learn course-related exercises with theoretical support. This kind of environment supports students in the learning process. This environment should be interactive and user-friendly.

The thesis consists of two parts: theoretical and practical. Analysis of the course and approaches to interactive learning environments are the focus of the theoretical part. The practical part of the thesis is creating a web application which meets the objectives stated before.

As a result, a web application for the subject Higher Mathematics was developed, which allows students to solve linear algebra exercises, what are supported by theoretical materials. A survey conducted among students found that the application is easy to use and offers real value in the learning process.

The thesis is in Estonian and contains 45 pages of text, chapters, 19 figures, 2 tables.

Lühendite ja mõistete sõnastik

AOI	<i>Action-Object-Input</i> , tegevus-objekt-sisend skeem interaktiivse õpikeskkonna loomisel
API	<i>Application Programm Interface</i> , rakendusliides
CLI	<i>Command Line Interface</i> , liides tekstipõhiste käskluste edastamiseks
DOM	<i>Document Object Model</i> , eeskiri objektide veebilehel esitlemiseks
EJS	<i>Embedded JavaScript templates</i> , mallimootor HTML mallide koostamiseks
HMAC SHA256	<i>HS256, Hash-based Message Authentication Code</i> , räsipõhine sõnumiautentimiskood, mis arvutatakse SHA265 räsifunktsiooni abil
HTML	<i>Hypertext Markup Language</i> , enimlevinud kodeerimissüsteem veebidokumentide loomiseks
HTTP	<i>Hypertext transfer Protocol</i> , andmevahetusprotokoll dokumentide vahetamiseks
JDBC	<i>Java Database Connectivity</i> , Java programmiliides andmebaasiga ühendumiseks
JSON	<i>JavaScript Object Notation</i> , andmevahetusvorming
JWT	<i>Json Web Token</i> , standard, mis võimaldab turvalisel viisil osapoolte vahel andmeid vahetada
MVC	<i>Model-view-controller</i> , arhitektuurimuster veebiteenuste kujundamiseks
NPM	<i>Node Package Manager</i> , keskkond JavaScript pakettidele
ORM	<i>Object-relational mapper</i> , võimaldab olemeid käsitleda objektorienteeritud keeltes
REST	<i>Representational State Transfer</i> , arhitektuuriline lahendus veebiteenuste arendamiseks
SCSS	<i>Sassy Cascading Style Sheet</i> , keel veebilehe kujundamiseks
SQL	<i>Structured Query Language</i> , struktureeritud päringute keel relatsioonandmebaaside kasutamiseks
XSS	<i>Cross site scripting</i> , koodisüsti meetod veebilehe ründamiseks

Sisukord

1 Sissejuhatus	10
2 Probleemi taust ja rakenduse eesmärk.....	11
2.1 Probleemi taust	11
2.2 Rakenduse eesmärk	12
2.3 Metoodika.....	12
2.4 Olemasolevad lahendused	13
2.5 Olemasolevate lahenduste kokkuvõte ja võrdlus.....	15
3 Lahenduse analüüs.....	17
3.1 Rakenduse nõuded	17
3.1.1 Funktsionaalsed nõuded	17
3.1.2 Mittefunktsionaalsed nõuded.....	18
3.2 Lineaaralgebra moodul kursuses Kõrgem matemaatika.....	18
3.3 Interaktiivse õpikeskkonna loomise meetodid	20
3.3.1 AOI lähenemisviisi sobitamine teostatavasse lahendusse	22
3.3.2 Abistava materjali kuvamise võimalused	22
4 Kasutatavad tehnoloogiad.....	24
4.1 Programmeerimiskeelte valikud	24
4.2 Klientrakenduse tehnoloogiate valikud	25
4.3 Serveripoolse rakenduse tehnoloogiate valikud	27
4.4 Andmebaasi valik	29
5 Lahenduse arendus	31
5.1 Kasutajaliidese disain	31
5.1.1 Kasutajavood rakenduses	31
5.1.2 Kasutajaliidese disain	33
5.2 Andmebaasi olemi-suhte diagramm	36
5.3 Serveripoolse rakenduse arendus.....	37
5.3.1 Spring Boot rakenduse loomine	37
5.3.2 Serveripoolse rakenduse arhitektuur	39

5.3.3 Andmebaasi loomine rakendusele	40
5.3.4 REST API.....	41
5.3.5 Spring Security turvalisus.....	44
5.4 Klientrakenduse arendus.....	45
5.4.1 Klientrakenduse loomine.....	45
5.4.2 Suhtlus serveripoolse rakendusega.....	47
5.4.3 Kasutajate ligipääs rakendusele.....	48
5.4.4 Kasutajaliidese tõlked.....	48
5.4.5 Mathlive teek	48
6 Tulemused	50
6.1 Loodud funktsionaalsused	50
6.2 Kasutajate tagasiside.....	51
6.3 Edasiarenduse võimalused.....	52
7 Kokkuvõte	54
Kasutatud kirjandus	55
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	59
Lisa 2 – Rakenduse vaadete prototüübid	60
Lisa 3 – Valminud rakenduse kuvatõmmised.....	62
Lisa 4 – Tagasiside küsimustik.....	64

Jooniste loetelu

Joonis 1. T-algebra rakenduse ülesande lahendamise vaade.....	21
Joonis 2. Interaktiivne ülesande lahendamise voog.....	23
Joonis 3. Mathlive veebikomponendi testimise tulemus.	26
Joonis 4. Kasutajavoog testide loomisel.....	32
Joonis 5. Kasutajavoog ülesannete ja testi lahendamisel.	33
Joonis 6. Ülesannete koondvaade prototüüp.	34
Joonis 7. Ülesande lahendamise vaade.....	35
Joonis 8. Ülesande lisamise vaade.....	35
Joonis 9. Andmebaasi olemi-suhte diagramm.....	36
Joonis 10. Loodava projekti põhi.	38
Joonis 11. Spring rakendusele lisatavad sõltuvused.....	38
Joonis 12. Serveripoolse rakenduse arhitektuur.	39
Joonis 13. Serveripoolse rakenduse kaustade struktuur.	40
Joonis 14. Andmebaasi migratsioonid rakenduse arenduse jooksul.	41
Joonis 15. JWT tokeniga autoriseerimise ja autentimise voog.....	44
Joonis 16. Klientrakenduse kaustade struktuur.	46
Joonis 17. Ühe mooduli kaustade struktuur.	46
Joonis 18. Koodinäide autentimise pealtkuulajast.....	47
Joonis 19. Hinnangute jaotus rakenduse kasutusmugavusele.	51

Tabelite loetelu

Tabel 1. Rakenduste funktsionaalsuste võrdlused.....	16
Tabel 2. REST API ressursid.....	43

1 Sissejuhatus

Kursus Kõrgem matemaatika on kohustuslik õppeaine kahele Tallinna Tehnikaülikooli IT Kolledži õppekavale: IT süsteemide arendus ja IT süsteemide administreerimine. Hetkel on kursuse õppematerjalid koondatud Moodle õppekeskkonda. Moodle platvorm pakub piisavalt paindlikku teoreetiliste testide koostamisvõimalust tudengitele [1]. Samas puudub aga paindlik viis, kuidas tudengitele teema paremaks omandamiseks harjutusülesandeid anda.

Lõputöö eesmärk on lahendada antud probleem läbi tarkvaraarenduse: luua veebirakendus, mis toetub kursuse materjalidele, aitab õpetatavat materjali paremini omandada ning võimaldab pakkuda suurt hulka interaktiivseid harjutusülesandeid. Lisaks peab rakendus võimaldama õppejõul korraldada hinnatavaid harjutusteste ja näha statistikat tudengite õppimisaktiivsuse kohta.

Lõputöö raames tutvutakse kursuse Kõrgem matemaatika ülesehitusega, uuritakse interaktiivse õppekeskkonna loomise meetodeid ning samuti ka vihjete ja teoreetiliste materjalide kuvamise viise, et valmiv lahendus toetaks tudengeid õppimisel võimalikult palju. Rakenduse loomisel arvestatakse, et õppimise juures on oluline pakkuda tudengitele ühest kohast harjutusvõimalust koos abistava materjaliga, et tekiks terviklik ülevaade lahendatavast probleemist. Teooria ei saa jääda rakendusest täielikult välja, vaid tuleks kuvada tudengitele ülesannete juures vihjetena ning lisaselgitustena interaktiivsel viisil.

Töö esimeses osas analüüsitakse täpsemalt probleemi ja selle tausta, seatakse lõputöö jaoks täpsemad eesmärgid ja skoop. Eesmärkide põhjal uuritakse olemasolevaid lahendusi, nende eeliseid ning ka puudusi, mida rakenduse arendamisel saab arvesse võtta. Järgmises osas analüüsitakse teostatavat lahendust, seejärel leitakse vastavalt analüüsile sobivad tehnoloogiad rakenduse arendamiseks. Eelviimases osas antakse ülevaade teostusest ning tekkinud probleemidest. Viimases osas kirjeldatakse tulemusi ning analüüsitakse kasutajate tagasisidet ja samuti ka võimalikke tehtud parandusi tagasiside põhjal.

2 Probleemi taust ja rakenduse eesmärk

Antud peatükis kirjeldatakse probleemi tausta, pannakse paika lahenduse eesmärgid ja lõputöö metoodika. Arvestades seatud eesmärke, analüüsitakse olemasolevaid keskkondi, nende vastavust eesmärkidele ning leitakse eelised ja puudused, mida uue lahenduse juures arvesse saaks võtta.

2.1 Probleemi taust

Kõrgem matemaatika on IT süsteemide arenduse ning IT süsteemide administreerimise õppekaval kohustuslik õppeaine. Kõrgema matemaatika kursuse raames käsitletakse kolme matemaatika valdkonda: matemaatiline analüüs, tõenäosusteooria ja matemaatiline statistika ning lineaaralgebra.

Matemaatika on küllaltki raske valdkond mitmetele tudengitele, eriti sessioonõppe tudengitele, kellel võib olla ligi 10 aastat viimasest matemaatika tunnist. Võttes arvesse matemaatiliste terminite keerukust, võib tudengitel olla raske iseseisvalt asjakohaseid täiendavaid materjale leida. Puudu on keskkond, kus tudengitel on võimalik leida kõrgema matemaatika kursusel läbitavate teemade kohta täiendavaid praktilisi ülesandeid ja lahendamisel (harjutamisel) õppida, kuidas jõutakse samm-sammult õige lõpplahenduseni.

Kuigi on olemas mitmeid rakendusi, mida on võimalik kasutada abivahendina kursuse läbimisel, on neil mitmeid puudusi. Raske on leida kvaliteetset õppematerjali, mis sobiks täielikult kokku kõrgema matemaatika kursuse raames käsitletud teemadega ning pakuks harjutamiseks interaktiivseid, toetava teoreetilise materjaliga ülesandeid. Olemasolevatest lahendustest kirjutatakse lähemalt peatükis 2.4.

Arvestades ka praegust distantsõppe olukorda, on kasulik, kui tudengitel on olemas erinevaid lisavõimalusi õppimiseks ning õpitava materjali kinnistamiseks. Samuti on tudengid nüüd rohkem harjunud veebipõhiste materjalidega ning võimaldades ülesandeid lahendada ka veebi vahendusel, on teemakohase materjali leidmine ja

teadmiste omandamine kindlasti lihtsam ja atraktiivsem. Loodav rakendus annaks hea võimaluse materjalide omandamiseks ja teadmiste kontrolliks ka distantsõppe lõpul.

Lahenduse skooopi jääb lineaaralgebra moodul, sest antud moodulit käsitletakse kõige viimasena ning seega annab võimaluse lõputöö raames rakendust testida hetkel kursust läbivate tudengitega.

2.2 Rakenduse eesmärk

Eesmärk on pakkuda tudengitele abivahendit, mis on võrreldes tavapärase lahendustega interaktiivsem, toetab materjalide paremat omandamist, pakkudes võimalust lahendada praktilisi ülesandeid, jagades tudengitele lahendamise käigus toetavaid ja terviklikke teoreetilisi näpunäiteid. Lahenduse eesmärk ei ole ära lahendada tudengite eest kodutöid.

Samuti on võimalik loodavas rakenduses õppejõul saada ülevaadet tudengite õppimisaktiivsuse kohta, et saada paremat tagasisidet tudengite vajaduste ja edasijõudmise kohta. Lisaks sellele luuakse rakendusse võimalus korraldada hinnatavaid teste, mille tulemusi saab õppejõud hindamisel arvesse võtta.

Lõputöö skoop piirdub vaid ühe mooduliga, kuid rakenduse edasiarendustel peab olema võimalik lisada ka teiste moodulite kohta sarnane (teoreetiline ja praktiline) tugi, et kogu kõrgema matemaatika kursuse harjutusmaterjal oleks tulevikus ühest kohast kättesaadav.

2.3 Metoodika

Esmalt kirjeldatakse probleemi ning seatakse töö jaoks täpsemad eesmärgid. Eesmärkide põhjal uuritakse olemasolevaid lahendusi, analüüsitakse nende puuduseid ning eeliseid, mida loodava lahenduse juures saab arvesse võtta.

Rakenduse funktsionaalsete ning mittefunktsionaalsete nõuete kaardistamiseks tehakse intervjuu õppejõuga, kes antud kursust tudengitele annab. Lisaks koostatakse nõuete ideede kogumiseks lühikene küsimustik, millele saavad vastata nii kursust sooritanud kui ka hetkel läbivad tudengid.

Lahenduse väljatöötamisel tuleb uurida kursuse ülesehitust ning paika panna strateegia ülesannete lahendamiseks. Samuti uuritakse interaktiivsete õppekeskkondade lähenemisviise, et valida õppimist toetav lähenemine rakenduse arendamiseks.

Rakenduse valmimisel tuleb hinnata rakenduse vastavust seatud eesmärkidele. Testida tudengitega nii rakenduse kasutegurit kui ka kasutusmugavust.

2.4 Olemasolevad lahendused

Olemas on mitmeid veebi- ja mobiilipõhiseid rakendusi kas veebikalkulaatori või tasuta kursuseid pakkuva keskkonna näol, mida on võimalik kasutada abivahendina kursuse läbimisel.

Kursused on kindla ülesehitusega, neid on nii tasulisi kui ka tasuta. Kursuste eeliseks on tervikliku materjali pakkumine: kasutajatele pakutakse üldjuhul nii teoreetilisi materjale kui ka praktilisi ülesandeid. Kuid puuduseks see, et olulise materjali leidmine võib tudengi jaoks muutuda liigselt keeruliseks või ülesandeid ei ole piisavalt.

Veebikalkulaatorid on oma ülesehituselt paindlikumad, sest tudeng saab soovitud probleemi trükkida kalkulaatorisse ja see lahendatakse tema eest ära. Mõne rakenduse puhul kuvatakse ka vahesamme ning lõpuks kuvatakse õige vastus. Selline lähenemine on õppimise eesmärgil parem vastuste kontrollimiseks kui teadmiste omandamiseks.

Järgnevalt analüüsitakse seitset olemasolevat lahendust, mis on seatud eesmärkidele lähedasemad.

T-algebra

T-algebra on eestikeelne põhikooli õpilastele suunatud interaktiivne õppekeskkond algebra ülesannete lahendamise õppimiseks ning ülesannete koostamiseks. Programm kontrollib lahenduskäigu õigsust igal sammul ning pakub tuge vihjete näol. Kuigi see rakendus ei ole lineaaralgebra ülesannete lahendamiseks mõeldud, on antud keskkonnas õppimisprotsess väga hästi lahendatud ning kontrollitud [2].

Khan Academy

Khan Academy on veebipõhine õppekeskond, mis pakub tasuta materjali õppimiseks mitmes erinevas valdkonnas [3]. Kursused on hästi struktureeritud ning sobivat teemat on kasutajatel lihtne leida. Keskkonna eeliseks on mitmekülgse teoreetilise materjali olemasolu ning puuduseks praktiliste ülesannete piiratud arv [4].

Udacity

Udacity on infotehnoloogia tudengitele suunatud õppekeskkond [5], mis õpetab lineaaralgebrat programmeerimise kaudu [6]. Selline lähenemine võib fookuse lineaaralgebra õppimisest eemale viia ning seetõttu ei ole harjutusülesannetest kasu kursuse Kõrgem matemaatika kontekstis.

Grasple

Grasple on ülikoolidele suunatud veebikursuste tegemise platvorm, mis on individuaalseks kasutamiseks tasuta, pakkudes materjale nii matemaatika kui ka statistika valdkonnast [7]. Lineaaralgebra ülesandeid on keskkonnas arvukalt ning erineva raskusastmetega, kuid samas puudub teoreetiline tugi [8]. Rakenduse eeliseks on kasutajale lahenduskäigu kuvamine peale õige vastuse sisestamist, mille abil on võimalik analüüsida lahenduskäigu õigsust.

Symbolab

Symbolab on veebipõhine kalkulaator, mida on võimalik kasutada abivahendina lineaaralgebra ülesannete lahendamisel [9]. Rakendus võimaldab kasutajal näha ülesannete lahenduskäike, kuid puudu on teoreetiline materjal. Rakendus sobiks kursuse kontekstis ülesannete vastuste kontrollimiseks.

IXL Learning

IXL Learning on õppekeskkond, mis on suunatud üldharidustasemele [10]. Rakendus pakub interaktiivseid ülesandeid, mis on toetatud asjakohase teooria ja lahendusnäidetega. Rakendus pakub õppimisel lineaaralgebra vallast maatriksitega seonduvat [11] ning sobiks harjutamiseks, kui maatriksitega seonduvad teemasid on raske mõista, kuid suuremat väljakutset rakendus ei paku.

Coursera

Coursera pakub veebikursuseid mitmetest erinevatest valdkondadest, samuti matemaatika ja ka lineaaralgebra vallast [12]. Kuna keskkonnas on nii tasuta kui ka tasulisi kursusi on keeruline leida sobiva teema jaoks vastavat kursust või materjale, et ülesandeid lahendada.

2.5 Olemasolevate lahenduste kokkuvõte ja võrdlus

Olemasolevate lahenduste hulgas ei olnud ühtegi rakendust, mis täidaks täielikult seatud eesmärgi. Läbivaim puudus oli vähene harjutusülesannete arv ning mõnel juhul piiratud või puuduv teoreetiline tugi. Harjutusülesanded olid nii avatud vastusega kui ka valikvastustena. Keerukamate ülesannete puhul on keeruline kõiki vaheetappe meelde jätta ning tuleb leida abivahend vahetehete tulemuste kirjapanemiseks. Ükski leitud lahendustest ei paku tasuta versioonis võimalust õppejõul saada tudengite õppimisaktiivsuse kohta statistikat.

Tabel 1 toob välja funktsionaalsused, mida võrreldi erinevate lahenduste puhul, ning nende olemasolu erinevates rakendustes. Märk „+“ tähendab funktsionaalsuse olemasolu rakenduses ning „-“ puudumist.

Tabel 1. Rakenduste funktsionaalsuste võrdlused.

Rakendus	Teoreetiline materjal	Harjutus-ülesanded	Kontroll-testid	Lahendus-käikude olemasolu	Tasuta ligipääs	Statistika tudengite aktiivsuse kohta
T-algebra	+	+	-	+	+	-
Khan Academy	+	+	-	+	+	-
Udacity	+	+	-	+	+	-
Grasple	-	+	-	+	+	+(tasulises)
Symbolab	-	-	-	+	+(osaliselt)	-
IXL Learning	+	+	-	+	+	-
Coursera	+	+	+	-	+(osaliselt)	-

Analüüsitud rakenduste eeliseid ja puuduseid võetakse arvesse järgmises peatükis tehtavas rakenduse analüüsis.

3 Lahenduse analüüs

Järgnevas peatükis pannakse paika rakenduse funktsionaalsed ja mittefunktsionaalsed nõuded. Analüüsitakse kursuse Kõrgem matemaatika sisu. Samuti uuritakse lähemalt interaktiivse õppekeskkonna loomise meetodeid.

3.1 Rakenduse nõuded

Rakenduse nõuded on tingimused või võimalused, mis kirjeldavad seda, kuidas probleemi saaks lahendada. Rakenduse nõuete määramisel on arvesse võetud autori kogemust ainet läbides, õppejõu soove, kaastudengite vajadusi ning ka uuritud olemasolevate lahenduste puuduseid ja eeliseid. Tudengitelt saadi vajaduste kohta informatsiooni *Google Forms* keskkonnas loodud küsitluse kaudu.

Rakenduse nõuded on koostatud eesmärgiga, et tekiks terviklik lahendus kursuse Kõrgem matemaatika jaoks, mis oleks lisaks ka mugava kasutajakogemusega. Nõuded on kirja pandud kolme tüüpi kasutajate (tudeng, õppejõud, administraator) põhjal.

3.1.1 Funktsionaalsed nõuded

Funktsionaalsed nõuded kirjeldavad seda, mida süsteem (rakendus) tegema peab. Sinna alla kuuluvad nii rakenduse ärioloogika toimimine kui ka kasutajate autentimine [13]. Järgnevalt on loetletud rakenduse funktsionaalsed nõuded:

- Ligipääs rakendusele peab olema autenditud. Tudengid peavad saama teha uut kasutajakontot ning rakenduses tehtud tegevused on seotud vaid selle kontoga. Rakendus peab kasutajatelt küsima konto loomiseks vaid minimaalset vajalikku infot.
- Rakenduses peab olema võimalik õppejõul ning administraatoril kasutajaid hallata ning nende ligipääsu rakendusele muuta.
- Õppejõud peab saama lisada praktilisi harjutusi, millest on võimalik teha hinnatav test. Lisatud ülesandeid saavad ka tudengid harjutamiseks kasutada.
- Õppejõul peab olema võimalik näha tudengite õppimisaktiivsuse statistikat.

- Rakendus peab koostama vastavalt ülesandetüübile harjutusülesandeid.
- Tudeng peab saama valida, milliseid ülesandeid lahendatakse.
- Harjutusrežiimis ülesannete lahendamine peab olema interaktiivne ehk rakendus peab pakkuma teoreetilist tuge ülesande lahendamiseks. Samuti peab olema võimalik kasutajal kirjutada rakendusse ülesande lahendamiseks vajalikke vahesamme mitte ainult vastust.
- Ülesannete lahendamiseks kasutatavad valemid peavad olema rakenduses kättesaadavad üldise loendina ning harjutusrežiimis ülesande juures.
- Harjutusrežiimis ülesannete lahendamisel peab rakendus pakkuma tagasisidet lahenduskäigu õigsuse kohta ning vea korral täpsustama vea kohta, et tudengil oleks võimalik tehtud vigu paremini analüüsida.

3.1.2 Mittefunktsionaalsed nõuded

Mittefunktsionaalsed nõuded kirjeldavad, kuidas süsteem (rakendus) toimib või käitub. Mittefunktsionaalsed nõudeid on erinevaid tüüpe, näiteks kättesaadavus, töökindlus ja kasutatavus [14]. Järgnevalt on loetletud rakenduse mittefunktsionaalsed nõuded:

- Rakendust peab saama kasutada väikeste ekraanidega seadmetel.
- Rakenduse kasutajaliides peab olema intuitiivne ning navigeerimine kasutaja jaoks selgesti mõistetav.
- Rakendus peab olema kasutatav enimlevinud brauserites.
- Rakenduse kasutajaliidese keel peab olema eesti keel ning koodibaasi keel inglise keel.

3.2 Lineaaralgebra moodul kursuses Kõrgem matemaatika

Lineaaralgebra on matemaatika valdkond, mis tegeleb lineaarfunktsioonidega, vektorite, vektorruumide ja maatriksitega [15]. Kursuse Kõrgem matemaatika raames käsitletakse lineaaralgebra põhiteemadena maatrikseid, determinante ning lineaarsete võrrandisüsteemide lahendamist. Kursuses on teemad jagatud neljaks selgeks osaks: vektorid ja maatriksid ning tehted nendega, determinandid, pöördmaatriks ja maatriksi astak ning lineaarsed võrrandisüsteemid ja nende lahendamisvõtted. Kursuse loengutes ning harjutustundides kaetakse lineaaralgebra põhitõed. Lineaaralgebrat on võimalik kasutusele võtta nii teoreetilises kui ka rakenduslikus matemaatikas kuid ka muudes

teadusvaldkondades, neist teemakohasemad informaatika, andmeanalüüs ning masinõpe [16, p. 26].

Lineaaralgebra mooduli esimeses osas kaetakse põhitehted maatriksitega. Need on samuti ka järgnevate keerukamate ülesannete (determinandid, pöördmaatriks, lineaarvõrrandisüsteemid) lahendamise aluseks:

- maatriksite liitmine ning lahutamine;
- maatriksi korrutamine skalaariga;
- maatriksi transponeerimine;
- rea- ja veeruvektorite skalaarkorrutus;
- maatriksite korrutamine.

Teises osas kaetakse determinandid:

- determinandi leidmine;
- maatriksi miinor;
- pöördmaatriks.

Kolmandas osas kaetakse maatriksvõrrandid:

- maatriksvõrrandisüsteemi lahendamine;
- maatriksi astaku leidmine;
- Kronecker-Capelli lineaarvõrrandisüsteemi lahendavuse teoreem;
- Crameri peajuht;
- maatriksite elementaarteisendused.

Neljandas ehk viimases osas kaetakse Gaussi elimineerimismeetod, mida kasutatakse lineaarvõrrandisüsteemi lahendamiseks.

Võttes arvesse lõputöö piiratud ajalist ja mahulist ressursi, on eesmärgiks pakkuda rakenduses harjutusvõimalust esimeses moodulis kaetud alateemade raames. Samuti õpetatakse nende alateemade raames teadmisi, mis on vajalikud järgmistes osades. Teised teemad jäävad esialgu edasiarenduste võimaluste nimekirja.

3.3 Interaktiivse õpikeskkonna loomise meetodid

Interaktiivseid õppekeskkondi on lähemalt uurinud Tartu Ülikooli arvutiteaduste instituudi informaatika kaasprofessor Marina Lepp. Tema sõnul on võimalik arvutipõhiste interaktiivsete õppekeskkondade lähenemised jagada kaheks: reeglitel (ingl. k *rule-based*) või käsklusel põhinevateks (ingl. k *command-based*) ja sisendipõhisteks keskkondadeks (ingl. k *input-based*) [17, p. 12].

Esimene ehk reeglipõhine keskkond baseerub „musta kasti“ lahendusel. Õppuril on võimalik valida kas reegel või avaldise osa, kuid teisendus tehakse programmi poolt. Must kast tähendab seda, et nähakse tulemust mitte tehte üksikasju. Seetõttu on lahendussammude õppimine passiivne, sest õppimiskeskond teeb rohkem õpilase eest ära, kui võimaldab ise nuputada [17, p. 12].

Teine ehk sisendipõhine keskkond tugineb paber-pliiats laadsel dialoogsüsteemil. Võimalik on kas järgida programmi algoritmi või sooritada ise algoritmi samm-sammult nii mitme etapiga kui vaja on. Sellega jõutakse olukorda, kus on võimalik leida vigu ainult lõpptulemusest, ja ei saa tuvastada, kus täpselt lahenduskäik valeks läks. Õppimise vaatest ei anna programm selliselt piisavalt tagasisidet [17, p. 13].

Esimese meetodi puuduseks on õppurite õppimisvõimaluse vähendamine ning teise puhul on raske hinnata seda, kust täpselt õpilasel lahenduskäik valeks läks. Seetõttu pakub Lepp koostöös teiste kolleegidega välja uue lähenemise interaktiivsetele õppimiskeskondadele: tegevus-objekt-sisend (edaspidi AOI, *Action-Object-Input*) lähenemine [17, p. 22]. AOI lähenemine ühendab kaks eelnevalt mainitud viisi. Igal sammul on lahenduskäigus kindel eesmärk [2]:

- teisendusoperatsiooni valimine sammuks (*action*);
- operandide valimine operatsiooniks (*object*);
- sammu tulemuse sisestamine ehk reegli rakendamine (*input*).

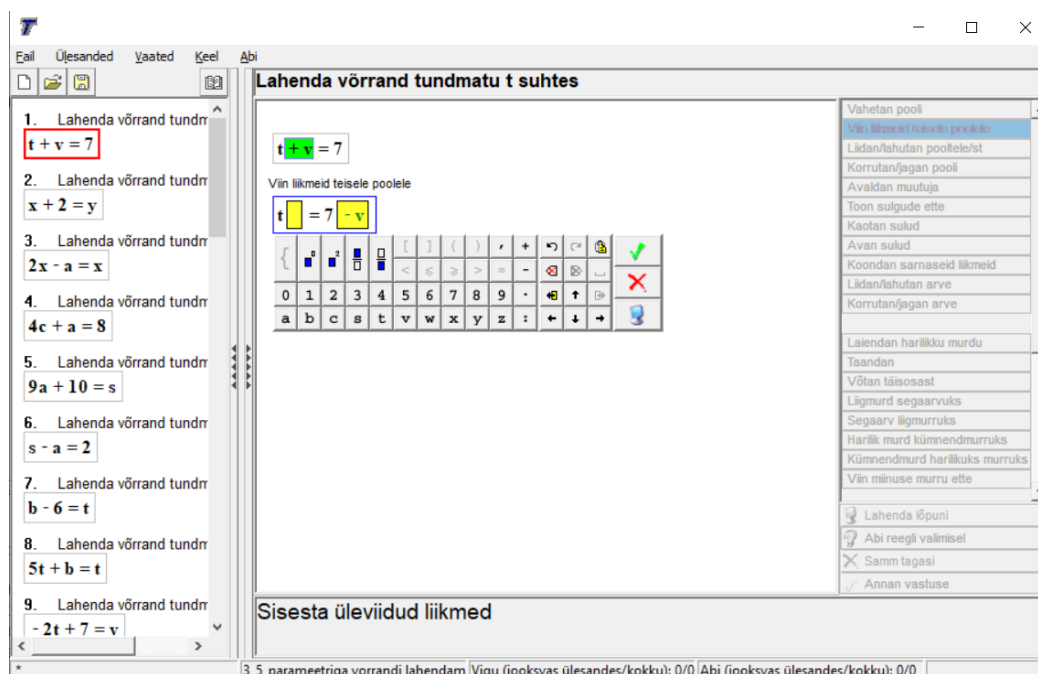
Antud skeemi esimest versiooni ehk *Action-Object* (AO, tegevus-objekt) lähenemist tutvustas Dmitri Lepp magistritöös, tuues välja, et ka ülesannete lahendamisel järgib lahendusmeetod (lahendusalgoritm) esmalt tegevust (lahendusreegel - *action*) ning järgmisena alles objekti (tehteosa - *object*), kuhu lahendusreeglit rakendada [18].

Marina Lepp ja kolleegid lisasid sellele viimase olulise osa: *input* ehk tulemuse sisestamine.

AOI lähenemist kasutati interaktiivse õppekeskkonna T-algebra arendamiseks. Sama lähenemisviisi plaanib ka käesoleva töö autor kasutada loodava rakenduse arendamisel, sest see on andnud märkimisväärseid tulemusi põhikooli õpilaste õppetulemuste paranemises [17].

T-algebra programmi üks parimaid õppimist võimaldavaid omadusi on selle „klaaskasti“ (ingl. k. *glass-box* ehk *white-box*) põhimõtetel ehitatud probleemilahendussüsteem [17, p. 32]. Erinevalt „musta kasti“ lahendusest on programmil võimalik aru saada lahenduse vahesammudest, nende õigsusest ning anda kasutajale tagasisidet lahenduskäigu kohta. Samas võimaldab see süsteem ka teha lahenduses vigu, mis on samuti õppimisprotsessis oluline.

Autor testis T-algebra rakendust, et saada ülevaadet rakenduse toimimise kohta ning kohandada paremini AOI lähfsamenemist lõputöö raames arendatava rakenduse jaoks. Joonis 1 näitab AOI skeemi viimast sammu: reegli rakendamine valitud kohale.



Joonis 1. T-algebra rakenduse ülesande lahendamise vaade.

Autor leiab, et antud rakendust oli küllaltki ebamugav kasutada võrrandi lahendamiseks, sest tuli teha mitmeid autori arvates ebavajalikke väikseid samme. Autori arvamus

olenemata on siiski hea, et on võimalik teha ka väga väikeseid samme, mis aitavad paremini algebrat või üldisemalt matemaatikat õppida. Plaanitavas lahenduses tahab autor võimaldada nii suuremate kui ka väiksemate sammude tegemise lahendamiskäigul.

3.3.1 AOI lähenemisviisi sobitamine teostatavasse lahendusse

Lineaaralgebra moodulis olevatel ning lahenduse skooopi jäävatel ülesannetel ei ole selliseid teisendusreegleid kui algebra ülesannetes, seega ei saa loodavas rakenduses *action*-i roll olla sama. Selle asemel on *action* lahenduskäigu vahesammu tegevus või otsus, mis vastab ülesandetüübi lahendusalgorithmile. Muutub ka *object*-i roll: see peab olema element, mida järgmiseks sammuks on vaja sisestada. *Input* peab võimaldama tudengil sisestada lahenduse järgmist sammu, mille õigsust tuleb rakendusel kontrollida. Sellest tulenevalt on võimalik liigitada AOI sammud loodava rakenduse kontekstis järgnevalt:

- *action*: soovitud tegevuse valik (järgmisena tehtav tehe jne.);
- *object*: järgmise sammu jaoks sisestatav element;
- *input*: sisendi kirjutamine vahesammu või lõpptulemuse jaoks.

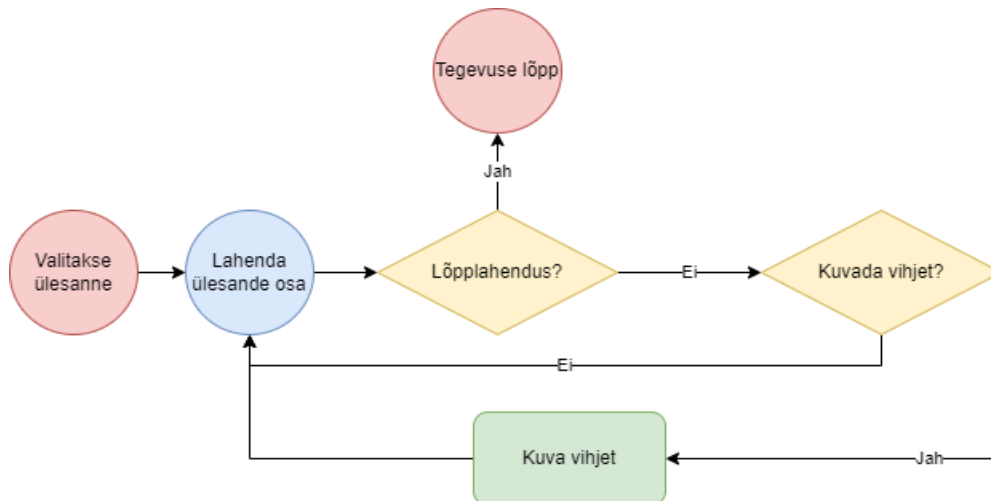
Antud lahendus võimaldab tudengil ülesande lahendamisel (harjutamisel) õppida, kuidas jõutakse samm-sammult lõpplahenduseni.

3.3.2 Abistava materjali kuvamise võimalused

Matemaatikas on teoorial lahenduskäikudel väga oluline osa, seetõttu tuleb õppimisel harjutamise käigus pakkuda teoreetilist abimaterjali. Abimaterjali pakkumiseks on kaks varianti: enne lahendama hakkamist või lahendamise käigus kasutaja soovil. Kaht eelmainitud vihjete pakkumise meetodit uuriti Razzaqi ja Heffernani poolt läbiviidud katsel põhikooliealiste seas. Katsetulemustest on võimalik järeldada, et pakkudes kasutajatele abimaterjali alles siis, kui soovitakse, on teadmiste omandamine parema tulemusega [19]. Lõputöö käigus valmivas rakenduses lähenetakse teoreetilise toe pakkumisel sarnaselt ehk siis kui kasutaja soovib ning seda interaktiivsel viisil.

Interaktiivsus informaatika kontekstis on suhtlemisprotsess inimese (kasutaja) ning arvuti (veebirakenduse) vahel. Enimlevinud kasutuskoht interaktiivsusel on mängudes,

kus kasutaja peab tegema edasi liikumiseks mingisuguse otsuse, mille järgi mäng teeb valiku ühes või teises suunas [20]. Sellist lähenemist on võimalik kasutada ka õppimise eesmärgil: Kasutaja valib ülesande ning hakkab seda lahendama, lahendamise käigus on võimalik kasutajal valida, kas soovib vihjet või lahendust, mida rakendus peab võimaldama (Joonis 2).



Joonis 2. Interaktiivne ülesande lahendamise voog.

Sarnast interaktiivset õppimismeetodit on kasutatud ka Carlos III Madridi ülikoolis inseneri eriala tudengitega, kus arendati kahe õppeaine jaoks rakendus, mis toetab tudengite paremat teemade omandamist arvuti- ning mobiilirakenduse abil. Tulemusena kasvas läbisaavate tudengite arv nelja aasta jooksul märgatavalt [21].

4 Kasutatavad tehnoloogiad

Töö tulemusena valmib veebirakendus, mis vastab peatükis 3.1.1 seatud nõuetele. Antud peatükis leitakse sobivamad tehnoloogiad lahenduse teostuseks.

4.1 Programmeerimiskeelte valikud

Rakenduse kliendi- ning serveripoolse raamistiku valimisel on kindad kriteeriumid. Nendeks on autori kogemus, programmeerimiskeel, vajalike teekide olemasolu ning populaarsus. Autori jaoks on oluline raamistikus kasutatav programmeerimiskeel kui ka kogemus antud tehnoloogiaga.

Iga raamistikuga kaasnevad kindlad programmeerimiskeeled ning autori jaoks on oluline, et kasutatav keel oleks staatiliselt ning tugevalt tüübitud. Staatiliselt tüübitud keelte puhul toimub tüübikontroll kompileerimise ajal, vastavad vead leitakse enne rakenduse käima minemist [22] ning lisaks sellele on kompileeritud koodi täitmine kiirem, sest kompilaator teab, milliseid andmetüüpe kasutada [23]. Tuntumad ning veebirakenduste arendamiseks sobivad staatiliselt tüübitud keeled on Java, C#, TypeScript.

Java ning C# on olemas olnud juba üle 20 aasta ning on tänapäevalgi suure kasutajaskonnaga keeled [24]. Mõlemal keelel on ka populaarsed veebirakenduse arendamiseks mõeldud raamistikud: Java puhul Spring [25] ning C# puhul ASP.NET [26]. Eesmärkide täitmiseks ning serveripoolse rakenduse arenduseks oleksid sobilikud mõlemad programmeerimiskeeled vastavate veebirakenduste raamistikega. Autoril on rohkem kogemust Java programmeerimiskeelega ning seetõttu välistatakse C# programmeerimiskeel.

TypeScript on eelnevalt mainitud keeltega võrreldes uuem, kuid siiski on aastatega selle populaarsus kasvanud, ning seda on võimalik kasutada nii klient- kui ka serveripoolses rakenduses [27]. Kuna autoril on varasem kogemus TypeScriptiga klientrakenduse arendamisel, siis valitakse see kliendipoolse rakenduse arendamiseks.

Et autor välistas programmeerimiskeelena C#, siis valikusse jääb Java serveripoolse rakenduse ja TypeScript klientrakenduse programmeerimiskeelena. Lähemalt kirjeldatakse serveripoolse raamistiku Spring võimalusi peatükis 4.3 ja võimalikke klientrakenduse raamistikke uuritakse peatükis 4.2.

4.2 Klientrakenduse tehnoloogiate valikud

Eelnevas peatükis mainiti kolme programmeerimiskeelt, mida on võimalik rakenduse arendamisel kasutada. Kuna autor välistas eelmises peatükis C# keele rakenduse arenduseks, siis jäävad valikusse vaid TypeScripti kasutatavad raamistikud.

Stack Overflow 2021 aastal läbiviidud küsimustiku andmetel moodustavad enim kasutatud veebiraamistike esineliku JavaScriptil või TypeScripti kasutatavad React.js, jQuery, Angular ja Vue.js [28]. Arvesse on võetud vaid kliendipoolse rakenduse arendamiseks mõeldud raamistikud ja teegid.

jQuery kasutab TypeScripti asemel JavaScripti. Võrreldes eelnevalt mainitud teiste raamistikega, nõuab jQuery rohkem koodi lihtsate tegevuste (näiteks komponendi seisu uuendamise) jaoks [29]. Nendel kahel põhjusel jQuery välistatakse.

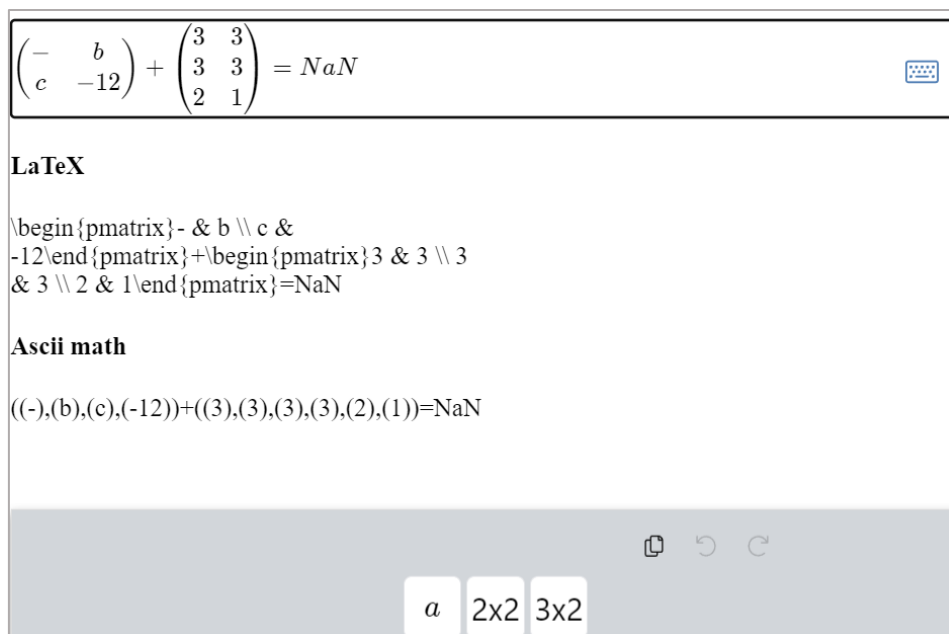
Alles jäänud kolmest raamistikust kasutab põhikeelena TypeScripti vaid Angulari raamistik, teised kasutavad JavaScripti, kuid neil on olemas ka TypeScripti tugi. Seega programmeerimiskeele vaatest on sobilikud kõik kolm teeki.

Ülesannete lahendamisel on oluline arusaadaval ja tuttavalt viisil kuvada kasutajale ülesannet ehk tehet. Antud juhul on vaja kuvada maatrikseid ning tavalisi matemaatilisi tehteid. Maatriksite puhul on enimlevinud kirjutusviisid kas kahe ümarsulu või nurksulu vahel eraldatud numbrid. Lisaks sellele on oluline ka sisestusvõimalus ehk kasutajal on võimalik ülesande lahenduse ajal kirjutada ise maatriks ja/või tavalisi tehteid ning seda võimalikult mugaval viisil. Nende kahe kriteeriumi alusel otsitakse ning võrreldakse erinevaid võimalikke teeki. Teekide otsimiseks kasutatakse nii Google otsingumootorit kui ka NPM (*Node Package Manager*) keskkonda. Autor leidis kolm võimalikku teeki, millega on võimalik tehteid kuvada. Nendeks on:

- MathJax [30];
- Mathlive [31];
- KaTeX [32].

MathJax on mallimootor (ingl. k *template engine*), KaTeX on JavaScripti teek ning Mathlive on veebikomponent. Veebikomponent on korduvkasutatav kohandatud komponent veebirakendustes kasutamiseks [33]. Kolmest teegist on TypeScriptis kirjutatud vaid Mathlive. Kõigi kolme teegi kuvamise loogika põhineb TeX või LaTeX keelel. LaTeX on märgistuskeel, mis põhineb TeX keele standardil [34]. Kuigi kõik kolm teeki võimaldavad LaTeX formaadis avaldiste kuvamist brauseris, on neist ainsana Mathlive-il olemas matemaatiliste tehete kirjutamiseks sisestusväli. Teise kahe teegi puhul tuleb sisestusvälja loogika ise juurde arendada.

Testimist alustati Mathlive-st, sest see oli kõige lähemal soovitud lahendusele. Selleks lõi autor uue Angulari rakenduse, kasutades raamistiku poolt pakutavat CLI (*Command Line Interface*) tööriista, ning lisas sellele vajaliku Mathlive sõltuvuse. Seejärel lisati sisestusväli (Joonis 3) tühja komponenti. Kuigi Mathlive on väga hästi dokumenteeritud, tekkis autoril raskusi selle integreerimisega Angulari rakendusse, sest dokumentatsioon on vaid suunatud HTML DOM manipulatsioonile mitte veebiraamistikuga kasutamisele.



Joonis 3. Mathlive veebikomponendi testimise tulemus.

Lisaks sisestusväljale on olemas ka virtuaalne klaviatuur, mis võimaldab luua ise nuppe ja vajalikke väärtusi, mida avaldise jaoks on vaja sisestada. Antud rakenduse puhul on oluline maatriksite sisestusvõimalus, mida tavalise klaviatuuriga teha ei saa. Autor lõi esialgu kaks nuppu, mis sisestavad maatrikseid, et näha, kuidas antud teek nende sisestamisega käitub. Tulemuseks oli mugav ja intuiitivne virtuaalne klaviatuur. Analüüsis mainitud AOI skeemi puhul vastaks klaviatuuril olev maatriksi valimise nupp objekt osale.

Et Mathlive täitis kõiki seatud eesmärged, otsustas autor jäädagi selle juurde ning teisi teke mitte testida. Samuti otsustas autor jääda klientrakenduse raamistikuga Angulari juurde, sest Mathlive ei seadnud piiranguid sobivale raamistikule ning autoril on vähem kogemust Reacti ning Vuega.

4.3 Serveripoolse rakenduse tehnoloogiate valikud

Autor välistas serveripoolse rakenduse programmeerimiskeelena C# ning TypeScripti varasemase vähese kogemuse tõttu. Programmeerimiskeeleks valiti Java, millega autor on varem kokku puutunud. Nagu peatükis 4.1 mainitud, on võimalik kasutada Spring raamistikku koos Java programmeerimiskeelega serveripoolse rakenduse arenduseks.

Spring raamistikuga arendamiseks on mitmeid eeliseid. Tegemist on ülemaailmselt kasutatava raamistikuga, millele on aastate jooksul tekkinud suur kasutajaskond ning seeläbi ka laialdane tugi. Samuti on Spring raamistikul olemas mooduleid, mida on vaja veebirakenduse arendamiseks kasutada. Nendeks mooduliteks on Spring JDBC (*Java Database Connectivity*) andmebaasiga suhtluseks, Spring MVC (*model-view-controller*) HTTP (*Hypertext Transfer Protocol*) päringute käitlemiseks, Spring Boot mugavamaks ja kiiremaks konfigureerimiseks ning Spring Security rakendusesisese autentimiseks ja autoriseerimiseks [35].

JDBC on tööriist andmebaasiga suhtlemiseks ning päringute tegemiseks. See võimaldab arendajal kirjutada andmete küsimiseks, lisamiseks ja muutmiseks SQL (*Structured Query language*) päringuid. Kuid sellest võib tekkida probleem: päringud võivad minna liialt keeruliseks ning arendajal tuleb pidevalt kontrollida nende süntaksi õigsust [36]. Alternatiivne võimalus JDBC-le on ORM (*Object Relational Mapper*). ORM võimaldab luua kihi relatsioonilise andmebaasi ja objekt-orienteeritud koodi vahel ilma, et andmete

kasutamiseks oleks vaja kirjutada SQL päringuid. See aitab suurendada produktiivust ning parandada rakenduse disaini. ORM-i puuduseks võib osutada jõudlus: genereeritakse liigselt keerulisi SQL päringuid [37]. Et ORM-i kasutamine lihtsustab arendust, võetakse ka lõputöö raames arendatava rakenduse jaoks see kasutusele. Enimlevinud ORM raamistik Java jaoks on Hibernate [38] ning seda kasutatakse ka antud rakenduse arendamisel.

Klientrakenduse ja serveripoolse rakenduse suhtlemiseks kasutatakse REST (*Representational State Transfer*) API (*Application Program Interface*) päringuid. See võimaldab kiiret ning mugavat andmevahetust HTTP abil ning Spring Boot sobib selleks väga hästi [39].

Lisaks veebirakenduse tehnoloogiale on oluline leida ka teek, mis lahendaks lineaaralgebra ülesandeid. Kuigi esialgu piirduakse vaid maatriksite elementaartehtetega, on hea, kui leitud teek võimaldab ka teisi ülesandeid lahendada. Leiti mitu sobivat teeki:

- Apache Commons Math [40];
- Colt [41];
- Weka [42]

Teekide puhul on oluline, et see võimaldaks teha kõiki elementaartehteid maatriksitega. Lisaks on ka hea, kui teek võimaldab teha ka muid tehteid, näiteks determinandi arvutamine, lineaarvõrrandi süsteemi lahendamine, maatriksi astaku leidmine ning Crameri peajuhu tingimuste kontroll.

Selgus, et kõik kolm teeki sisaldavad rakenduse arenduseks vajalikke funktsionaalsusi ning lisaks veelgi teisi matemaatilisi funktsionaalsusi. Seega selle järgi ei ole võimalik otsust teha ning vaadatakse veel seda, millal viimati teeki uuendati ning palju seda kasutatakse.

MVNRepository andmetel on Colt-i viimane versioon üles laetud aastal 2010, Apache Commons Math 2016 ning Weka 2022. Sellel võib olla mitmeid põhjuseid: matemaatikas üldjuhul reeglid ei muutu ning aja jooksul parandamist ei vaja, vajalikud funktsionaalsused on arendatud ja uute funktsionaalsuste vajadust ei ole tekkinud. Colt-i kasutatakse kokku 194 *artifact*'is, Weka-t 66-s ning Apache Commons Math-i 2094

artifact'is [40], [41], [42]. Sisemiselt kasutavad ka teised Apache teegid Apache Commons Math-i. Sellest võib järeldada, et tegemist on hästi toetatud teegiga. Seetõttu kasutatakse arenduseks Apache Commons Math teeki.

4.4 Andmebaasi valik

Andmebaase on võimalik liigitada mitmetesse erinevatesse kategooriatesse: relatsioonilisteks, hierarhilisteks, objektorienteerituks. Veebirakendustes kasutatakse kõige rohkem relatsioonilisi, seetõttu uurib autor vaid seda tüüpi andmebaase. Stack Overflow arendajate küsitluse tulemusena selgus, et enim kasutatud andmebaaside esikolmiku moodustab MySQL, PostgreSQL ja SQLite [43].

MySQL on Stack Overflow andmetel kõige populaarseim relatsiooniline andmebaas. Tegemist on laialdaselt kasutatavata andmebaasiga, mille arendusel keskenduti kiirusele ja töökindlusele. Kuid sellest tuleneb ka selle puudus: osad SQL võimalused on jäetud välja, et muuta andmebaasimootorit kiireks (näiteks *full-join*). Tegemist Oracle poolt hallatava andmebaasimootoriga ning selle edasiarendamine on muutunud küllaltki aeglaseks [44]. MySQL-ist nii tasuta kui ka lisadega tasuline versioon. Rakenduse arenduseks tõenäoliselt piisaks MySQL-i tasuta versioonist.

PostgreSQL on väidetavalt kõige arenenum avatud lähtekoodiga relatsiooniline andmebaas. See jälgib tähelepanelikult SQL standardit ehk toetab pea kõiki SQL-i võimalusi. Tegemist on objekt-relatsioonilise andmebaasiga, mis võimaldab kasutada keerukaid andmetüüpe nagu JSON (*JavaScript Object Notation*), mida MySQL ega SQLite ei võimalda. PostgreSQL-i arendajate kommuun on väga aktiivne ning pidevalt arendatakse uusi funktsionaalsusi [44].

SQLite on serverita andmebaas, mis salvestab andmed kohalikku masinasse ning seetõttu on andmetele ligipääs küllaltki kiire. Kuid samas ei paku SQLite mitmeid samaaegseid päringuid, mis osutub probleemiks kui kasutajaid on korraga rohkem kui üks. SQLite sobiks antud rakenduses vaid arenduse ajaks [44].

Lõplik valik osutus PostgreSQL-i kasuks, sest see jälgib täpsemalt SQL standardit, on täielikult avatud lähtekoodiga, seda arendab edasi aktiivne arenduskommuun ning kõiki

funktsionaalsusi on võimalik tasuta kasutada. Lisaks on autoril rohkem kogemust PostgreSQL-iga kui MySQL-iga.

5 Lahenduse arendus

Eelmises peatükis kirjeldati tehnoloogiad, mida arenduseks kasutatakse. Käesolev peatükk annab ülevaate rakenduse disainist ja teostamisest, võimalikest tekkinud probleemidest ja nende lahendustest.

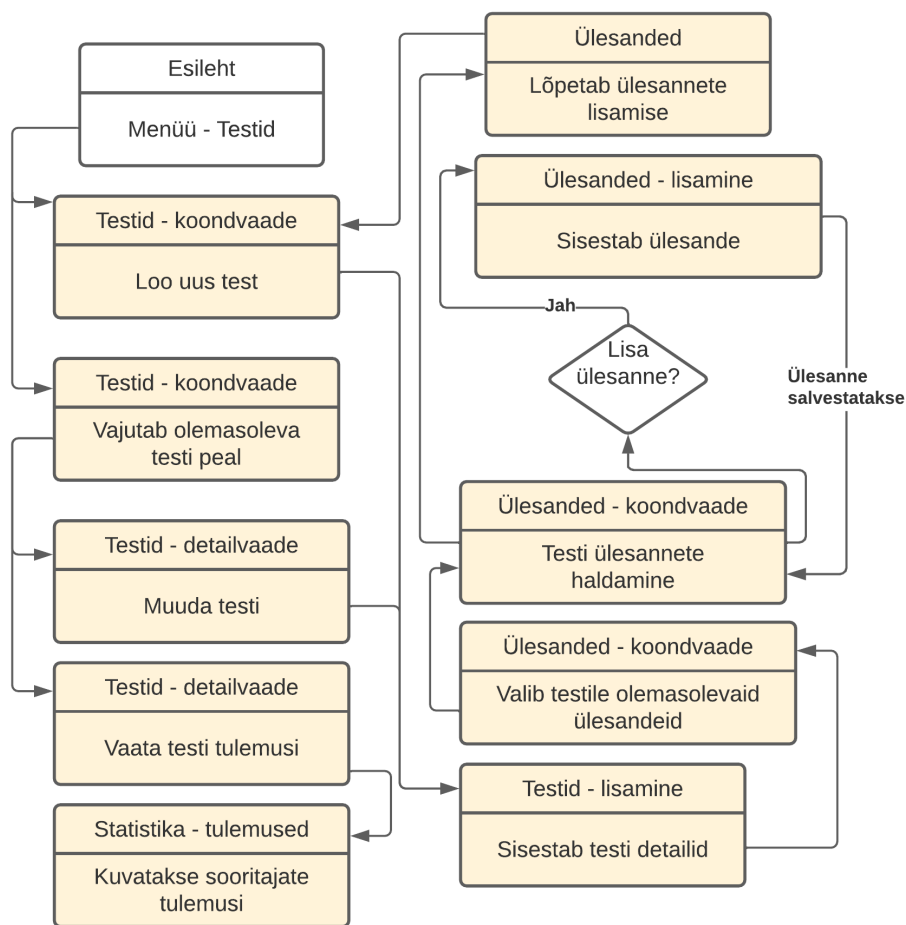
5.1 Kasutajaliidese disain

Enne veebirakenduse arendamise alustamist, tuleb teha kasutajavoo joonised ning klientrakenduse disain. Selline lähenemine aitab paremini hinnata seda, milliseid vaateid on veebirakenduses vaja.

5.1.1 Kasutajavood rakenduses

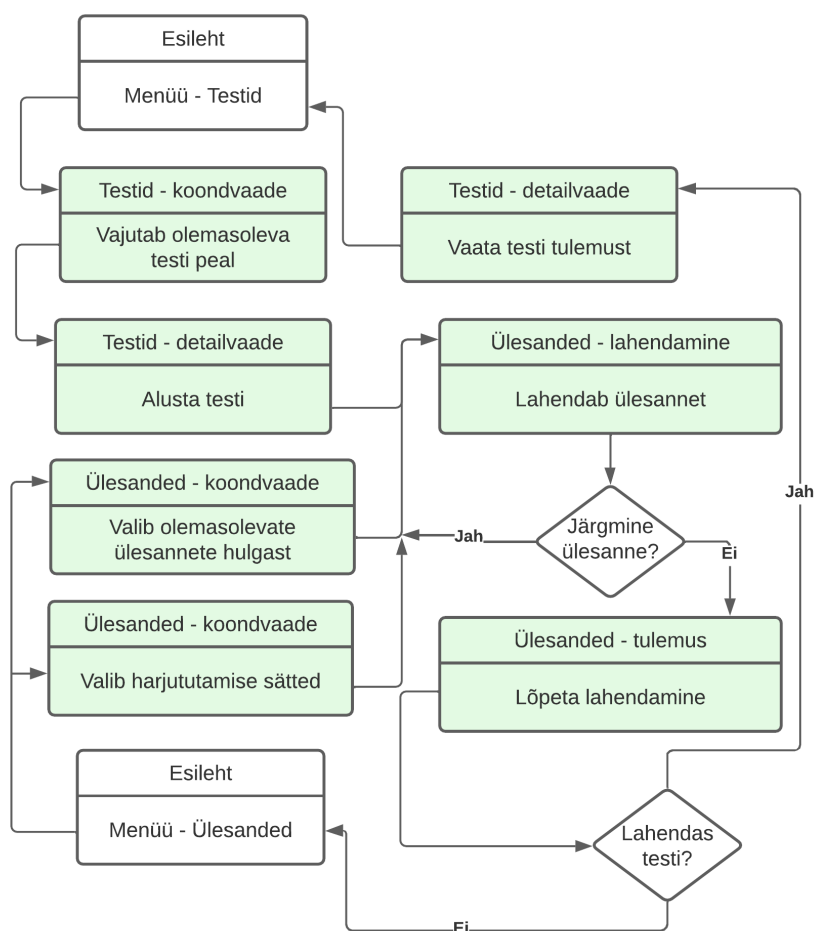
Rakenduses on kaks põhilist kasutajavoogu: tudengi rollis kasutajatele harjutamise ja testi lahendamise voog ning õppejõu rollis kasutajale testi loomise kasutajavoog. Kasutajavood läbivad mitmeid erinevaid vaateid, mida rakenduses on vaja, et soovitud tegevust võimalikult mugavalt ja selgelt teha.

Joonis 4 toob välja kogu voo, mida tuleb läbida, et luua keskkonnas test, mida on võimalik tudengi rollis kasutajatel lahendada. Põhilehelt on võimalik navigeerida testide vaatesse, kus saab hallata nii olemasolevaid teste kui ka lisada uusi teste. Testid koosnevad harjutusülesannetest. Neid on võimalik kas varasemaselt loodud testidest valida või ka uusi lisada. Olemasolevate testide juures on ka statistika vaade, kus saab näha testi lahendamise tulemusi.



Joonis 4. Kasutajavoog testide loomisel.

Ülesannete lahendamine eraldiseisvana ja testi osana on osaliselt kattuvad protsessid: vahe on sisuliselt vaid teoreetilise toe olemasolul (Joonis 5). Harjutamise alustamiseks on kaks võimalust: lahendatakse kas olemasolevaid ülesandeid või lastakse rakendusel genereerida juhuslikke ülesandeid. Valikule järgneb lahendusvaade, kus kogu õppimine sisuliselt toimubki. Testi puhul jõutakse samasse vaatesse, kuid teisel meetodil: enne avatakse õppejõu poolt loodud test ja seejärel ülesanded. Lõpp-punktiks on mõlemal vool koondvaade, vastavalt ülesannete või testide oma.



Joonis 5. Kasutajavoog ülesannete ja testi lahendamisel.

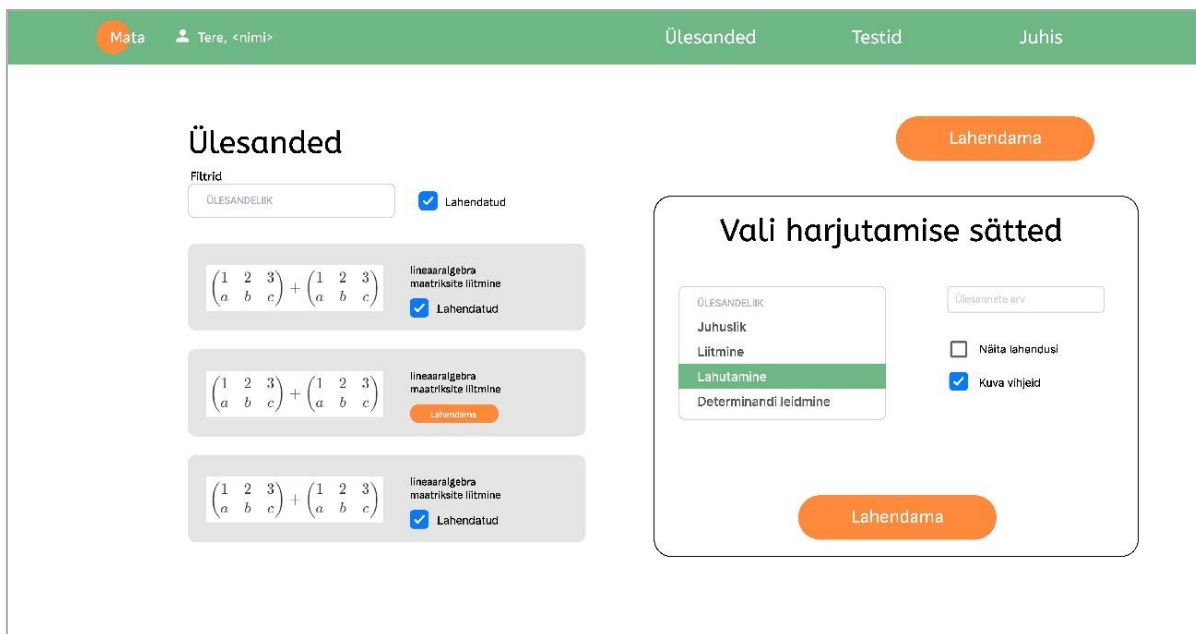
Mõlemad kasutajavood on loodud mõttega, et tegevuste sooritamine toimuks minimaalse vaevaga ning nendest oleks võimalik jõuda tagasi õigesse alguspunkti.

5.1.2 Kasutajaliidese disain

Kasutajavoogude põhjal selgus et rakenduses on neli põhivaadet: esileht, testid, ülesanded ja statistika. Selleks, et saada kasutajaliidestest visuaalset ettekujutlust, kasutab autor prototüüpimistööriista Figma [45]. See võimaldab luua eri mõõtmetes vaateid, mis aitab saada paremat ülevaadet teostatavatest funktsionaalsustest ja lehe elementide paigutustest, et rakenduse kasutamine oleks kasutaja jaoks intuiitiivne.

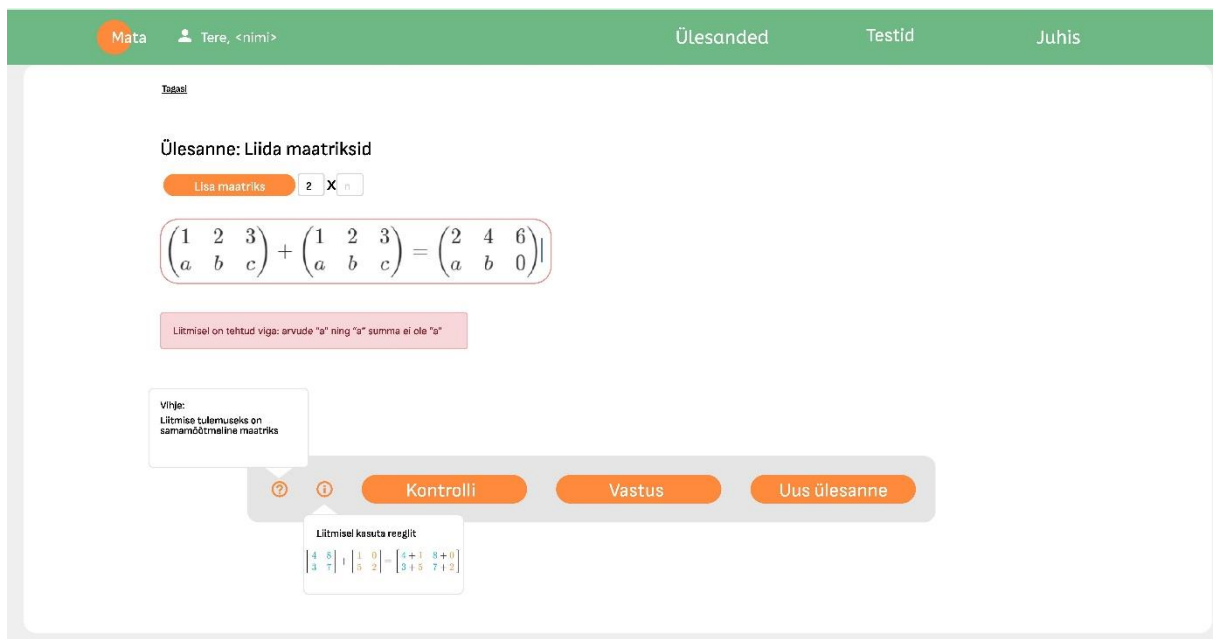
Prototüüpide disainimisel valis autor veebirakenduse põhivärvideks oranži ning roheline. Nendel kahel värvil on psühholoogiline efekt kasutaja käitumisel. Oranž ja ka muud soojad toonid soodustavad keskendumisvõimet. Roheline aga seostub enim rahulikkusega, õnnelikkusega ja elevusega [46].

Joonis 6 kujutab tudengi rollis kasutaja ülesannete koondvaadet. Siit on võimalik asuda lahendama nii olemasolevaid ülesandeid kui ka uusi ülesandeid. Lahendama minnes avatakse modaal (joonisel osa „Vali harjutamise sätted“), kus on võimalik valida sätteid ülesannete lahendamiseks.



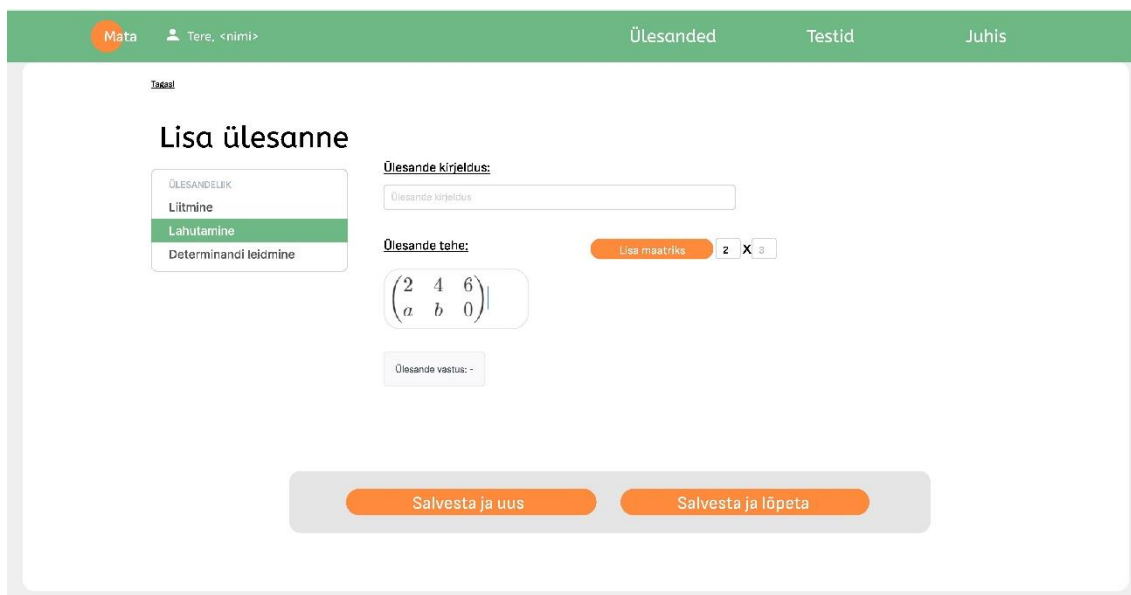
Joonis 6. Ülesannete koondvaade prototüüp.

Ülesande avamisel avaneb ülesande tekst koos tehtega, mida tuleb lahendama hakata. Harjutamisel on oluline see, et kontrollitaks igal vahesammul lahenduskäigu õigust ning kui on tekkinud viga, kuvatakse täpsema infoga teade (Joonis 7). Lisaks on olemas ka juhtpaneel, kust on võimalik leida vihjed, valemid ja muud vajalikku. Vihjeid ja valemeid kuvatakse tööriistavihjetena.



Joonis 7. Ülesande lahendamise vaade.

Samuti on võimalik paralleelsele tuua ka õppejõu rolli vaatest. Näiteks ülesande lahendamise asemel avaneb sarnane vaade ka ülesande lisamisel. Joonisel (Joonis 8) on näha lisaks avaldise sisestusväljale ka ülesande kirjeldus, liik ning teiste tegevustega juhtpaneel.

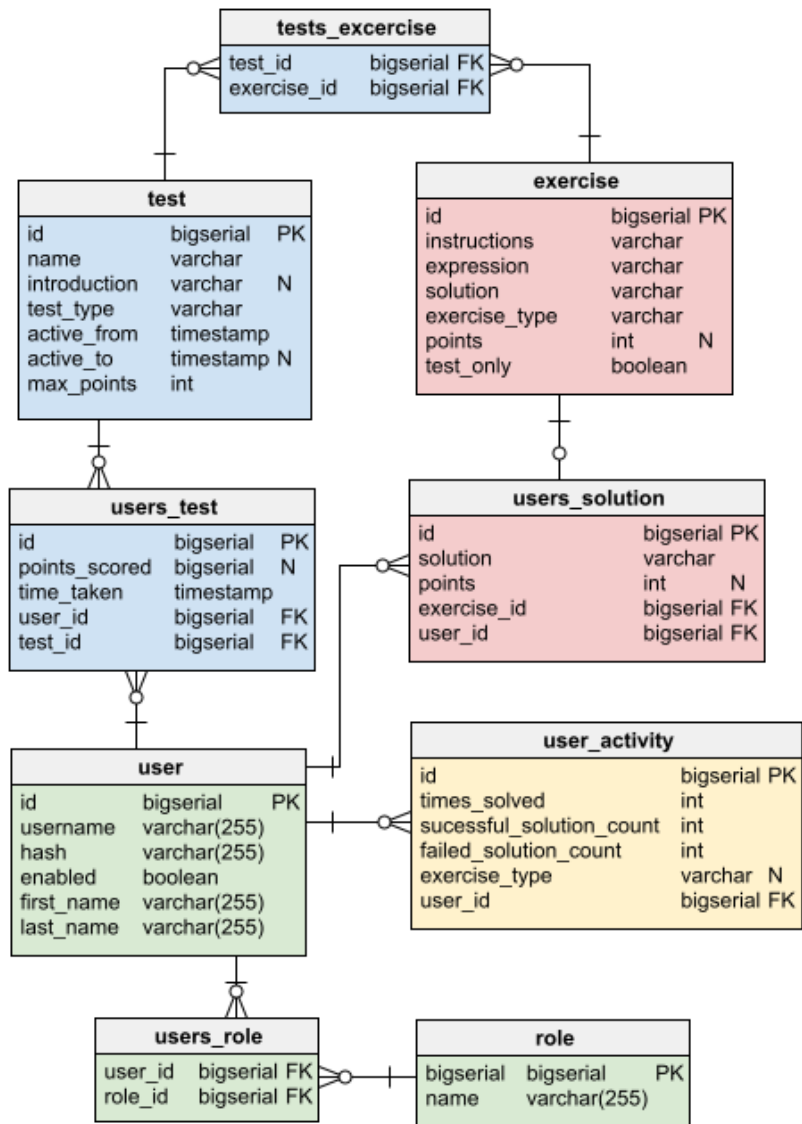


Joonis 8. Ülesande lisamise vaade.

Autor tegi ka teistele põhilisematele vaadetele prototüübid, millega on võimalik tutvuda lisades (Lisa 2).

5.2 Andmebaasi olemi-suhte diagramm

Funktsionaalsete nõuete, kasutajavoogude ja klientrakenduse prototüüpide põhjal on välja kujunenud, milline võiks olla antud rakenduse andmebaasi skeem. Et joonistada vastav olemi-suhte diagramm, kasutas autor Vertabelo keskkonda [47]. Kokku leiti, et vaja on üheksat tabelit (Joonis 9).



Joonis 9. Andmebaasi olemi-suhte diagramm.

Igal tabelil on andmebaasis oma roll:

- *User, users_role, role* – Spring Security raamistiku jaoks kasutajate ja nende õiguste ning rollide info talletamiseks.
- *Test* – õppejõu koostatud testide üldinfo hoidmiseks.
- *Exercise* – info õppejõu poolt lisatud ülesannete kohta.
- *Users_solution* – info kasutaja esitatud vastuse kohta.
- *Users_test* – tudengi sidumine lahendatud testiga ning info testi tulemuse kohta.
- *Tests_exercise* – ülesande sidumine lisatud testi juurde.
- *User_activity* – tabel õppimisaktiivsuse statistika jaoks, kogub infot nii salvestatud kui ka salvestamata ülesannete kohta.

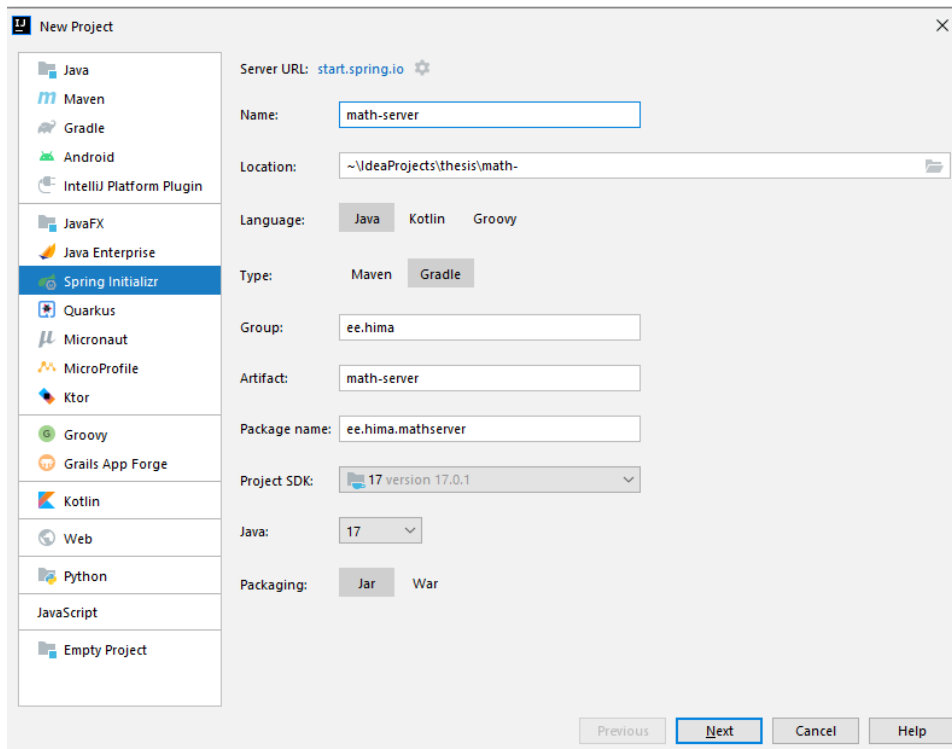
Joonisel on erinevat ülesannet täitvad tabelid märgitud eri värvidega. Rohelise taustaga on kasutajate autoriseerimise ja autentimisega seonduv. Sinise taustaga on hinnatavate testide loomise ja lahendamise seotud tabelid. Punase taustaga on ülesannetega seotud tabelid ning kollasega statistikaga seotud tabel. Terviklikuna võimaldab selline skeem luua õppejõul hinnatavaid teste koos harjutusülesannetega ning neid anda tudengitele lahendamiseks. Õppejõul on võimalik näha õppimisaktiivsust tudengite kohta ehk kui palju milliseid ülesandeid lahendatakse ning kui edukad ollakse.

5.3 Serveripoolse rakenduse arendus

Järgnevates alapeatükkides antakse ülevaade rakenduse loomise kohta, selle arhitektuuri ning äriloogika arendamise kohta. Serveripoolses rakenduses toimub andmetega suhtlemine ning nende vahendamine klientrakendusega HTTP abil.

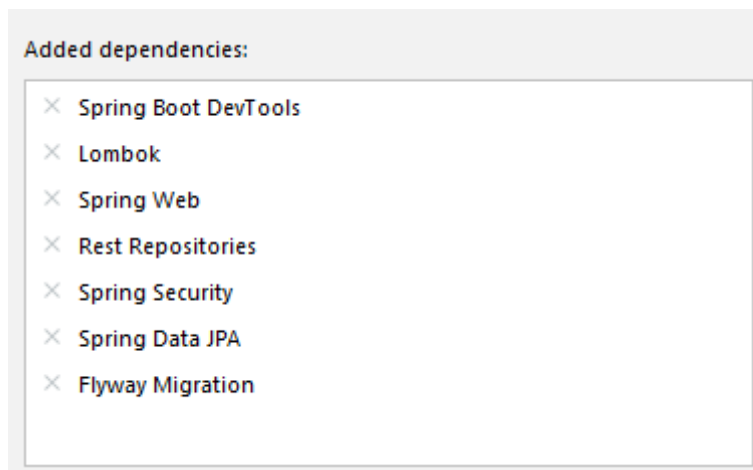
5.3.1 Spring Boot rakenduse loomine

Rakenduse loomiseks kasutati Intellij keskkonda integreeritud Spring Initializr tööriista, mille abil luuakse soovitud sõltuvustega tühi projekt. Joonis 10 näitab projekti põhja. Projektis valiti ehitustööriistaks Gradle ning Java versiooniks 17, mis on viimane pikaajalise toega Java versioon.



Joonis 10. Loodava projekti põhi.

Üheks eeliseks Spring Initializr-i kasutamiseks veebirakenduse loomisel on see, et on võimalik alguses esimesed teadaolevalt vajalikud sõltuvused projekti lisada (Joonis 11).



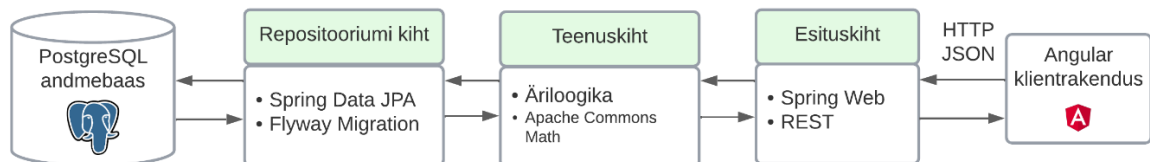
Joonis 11. Spring rakendusele lisatavad sõltuvused.

Spring Web on veebirakenduse tegemiseks, Rest Repositories võimaldab kasutada andmebaasiobjekte päringutes, Spring Security on autentimise ja autoriseerimise jaoks, Spring Data JPA on andmebaasiga suhtlemiseks ja Flyway Migration on andmebaasi migratsioonide tegemiseks ja haldamiseks.

5.3.2 Serveripoolse rakenduse arhitektuur

Mitmehilise arhitektuuriga rakenduste loomisel on mitmeid eeliseid. See aitab paremini struktureerida koodi ning anda igale kihile kindel loogiline ülesanne. Selline lähenemine aitab ka edaspidi paremini hallata koodi, täiendusi ning parandusi, sest kihid on üksteisest sõltumatud [48]. Sellise lahenduse loomiseks jagatakse koodibaas kolmeks koodikihtiks ning lisatakse ka neljas andmebaasikiht (Joonis 12):

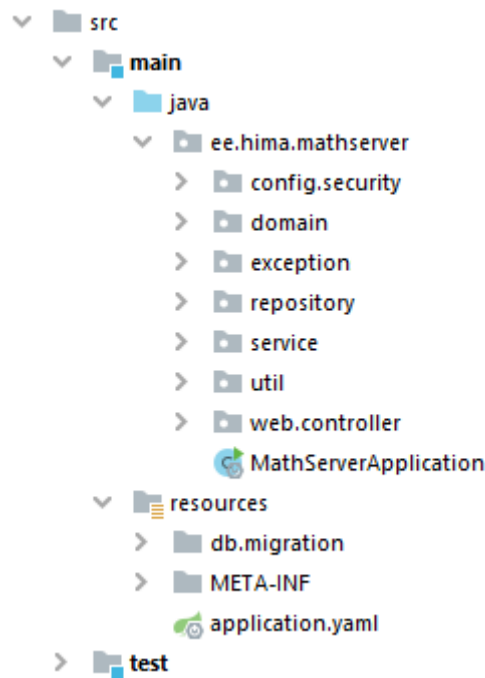
1. Esituskiht: API kontrolleri kiht, mille läbi suhtleb serveripoolne rakendus klientrakendustega ja vastupidi HTTP päringute abil.
2. Teenuskiht: äri loogika kiht, kus eksisteerib kogu rakendusega seonduv loogika.
3. Repositooriumi kiht: andmebaasiga suhtlemise meetodite kiht.
4. Andmebaasikiht: andmebaas koos salvestatud andmetega.



Joonis 12. Serveripoolse rakenduse arhitektuur.

Spring raamistik toetab sellise arhitektuuri loomist, sest selles eksisteerib kolm olulist komponenti, mida on võimalik vastavate annotatsioonide abil tähistada: kontrollerid (*@RestController*), teenused (*@Service*) ja repositooriumid (*@Repository*). Annotatsioonid tuleb märkida iga vastava kihi klassile, et raamistik võimaldaks sõltuvuste süstimise meetodi (ingl. k. *dependency injection*) abil komponentide omavahelist suhtlust. Meetod tähendab seda, et Spring raamistik lisab ise vajalikud sõltuvused ilma, et neid tuleks ise luua.

Sellise arhitektuuri järgi on võimalik panna paika ka serveripoolse rakenduse kaustade struktuur, mis jälgib lähedalt eelmainitud kihilist arhitektuuri. Iga kiht on jagatud eraldi kausta (Joonis 13). Lisaks on eraldatud repositooriumikihist andmemudeli objektid *domain* kausta ning rakenduse konfigureerimisega seonduv eraldi *config* kausta.



Joonis 13. Serveripoolse rakenduse kaustade struktuur.

Selline kaustade struktuur võimaldab hoida koodibaasis iga kihi loogikat eraldatuna ning selgelt nimetatud kaustad võimaldavad vajalikke kohti paremini üles leida. Lisaks on loodud eraldi alamkaustad ka erindite (ingl. k *exception*) ja abimeetodite (ingl. k. *utilities*, lühendatult *util*) jaoks.

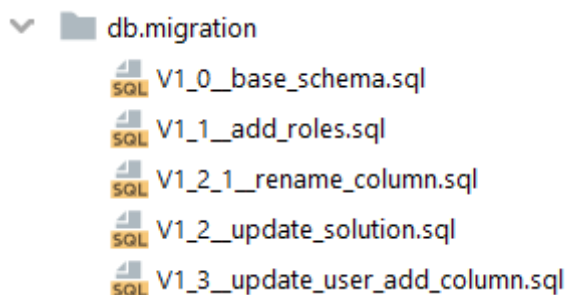
5.3.3 Andmebaasi loomine rakendusele

Andmebaasimootoriks valiti PostgreSQL kõige uuem võimalik versioon. Sinna loodi andmebaas, kuhu salvestatakse rakenduse töö käigus tekkinud andmeid. Andmebaasiga ühendumiseks defineeriti konfiguratsioonifailis parameetrid (andmebaasidraiver, andmebaasi võrguaadress ning andmebaasi kasutajanimi ja parool). Seejärel tehti andmebaasi tabelid.

Tabeleid on võimalik luua mitmel viisil. Üheks võimaluseks on kasutada Spring Data JPA-d, mis võimaldab annotatsioonide abil märkida klasse andmebaasiobjektideks *@Entity* annotatsiooni abil, märkida sellele vastav tabel andmebaasis *@Table* annotatsiooni abil ning samuti märkida ka olemite vahelisi suhteid (näiteks *@OneToMany* märgib üks-mitmele suhet olemite vahel).

Sellega aga kaasnevad andmebaasi versioneerimise probleemid, sest on küllaltki tõenäoline, et aja jooksul on andmebaasiskeemi vaja muuta. Muutes vaid koodis olevaid objekte, ei teki piisavat ülevaadet andmebaasialoost ning seetõttu tuleks andmebaasi versioneerida.

Andmebaasi migratsioonide jaoks kasutatakse Flyway tööriista. Sellega on võimalik luua migratsiooniskripte, mida on võimalik käivitada nii käsurealt kui ka sätestada skriptide automaatset täide saatmist enne rakenduse tööle minekut. Tööriista kasutamise eeliseks on andmebaasi versioonide ajaloo säilimine eraldi skriptifailides. Seeläbi tekib parem ülevaade andmebaasi eelmistest versioonidest ning on lihtne minna tagasi varasemase versiooni juurde [49]. Lisaks on võimalik autoril veenduda, et luuakse vaid sellised tabelid, mida vaja on. Migratsioonifailide nimetamisel tuleb järgida Flyway tööriista poolt määratud nimereeglit, mille järgi on versiooninumbri ja skripti nime vahel kaks alakriipsu (Joonis 14).



Joonis 14. Andmebaasi migratsioonid rakenduse arenduse jooksul.

Kuigi andmebaasi olemid luuakse migratsioonide abil, kasutab autor ära teist Spring Data JPA-ga kaasnevat mugavusfunktsionaalsust: andmebaasi valideerimine. Selleks konfigureeritakse vastav parameeter konfiguratsioonifailis *application.yaml*, mis igal rakenduse käivitamisel kontrollib andmebaasi tabelite vastavust koodis loodud mudelobjektidele. See aitab rakenduse arendamisel andmebaasi ning mudelobjekte hoida asjakohasena, et vältida erinevusi kahe eelmainitu vahel.

5.3.4 REST API

REST API võimaldab serverrakendusel suhelda läbi HTTP kliendipoolse rakendusega. API on kogum reeglitest, kuidas rakendused saavad omavahel suhelda, ning REST on arhitektuuriline lahendus mis põhineb kuuel piirangul, mida tuleks veebiteenuste

arendamisel kasutada. Näiteks peaksid ühe ressursi päringu tulemused olema samad olenemata sellest, kust päring ressursini jõudis [50].

REST API päringute ressursside aluseks võeti funktsionaalsed nõuded ning loodud kasutajavoogude joonised. Tabel 2 toob välja kõige olulisemad funktsionaalsustega seotud päringud, välja on jäetud kasutajate- ja rollihaldusega seotud ressursside kirjeldused.

Tabel 2. REST API ressursid.

Ressurs	HTTP meetod	Kirjeldus
/exercises	GET	Tagastab kõik rakenduses olevad ülesanded.
/exercises	POST	Lisab uue ülesande lahendamiseks.
/exercises/{id}	GET	Tagastab ID järgi ülesande.
/exercises/check	POST	Kontrollib esitatud lahenduskäigu õigsust.
/exercises/practice	POST	Tagastab vastavalt valitud lahendamissätetele ülesandeid lahendamiseks.
/exercises/solution	POST	Tagastab lisatava ülesande vastuse.
/exercises/solve/{id}	GET	Tagastab ID järgi ülesande lahendamiseks.
/statistics	GET	Tagastab üldise ülevaate õppimisaktiivsusest.
/statistics?type={exerciseType}	GET	Tagastab ülevaate õppimisaktiivsuse kohta ülesandetüübi järgi.
/tests	GET	Tagastab kõik rakenduses olevad testid.
/tests	POST	Lisab uue testi.
/tests/{id}	GET	Tagastab ID järgi testi.
/tests/{id}/exercises	GET	Tagastab kõik testis olevad ülesanded.
/tests/{id}/exercises/exlcuded	GET	Tagastab olemasolevad ülesanded, mis ei ole testis.
/tests/{id}/result/ /tests/{id}/result/{userId}	GET	Tagastab kasutajapõhiselt testi tulemuse.
/tests/{id}/result/all	GET	Tagastab ID järgi testi kõik tulemused.
/tests/{id}finish	POST	Kontrollib testi tulemust.
/tests/add/exercises	PATCH	Lisab testile ülesandeid.

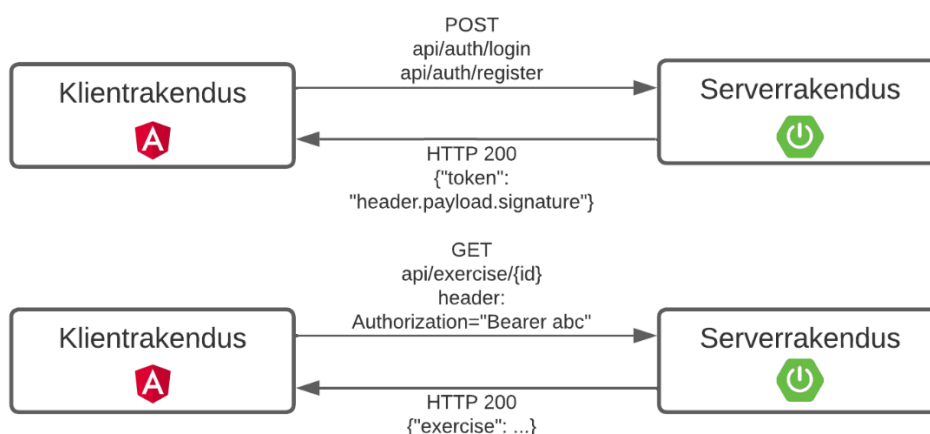
Ressursi nime kirjeldamiseks on kasutatud sellele vastava andmebaasi olemi nime, sest see kirjeldab kõige paremini selle funktsionaalsust. Iga päringu ette käib ka eesliide */api*, mis viitab sellele, et tegemist on API liidesega. Päringute juures on kirjeldatud vastava ressursiga seonduv HTTP meetod (GET, POST või PATCH).

5.3.5 Spring Security turvalisus

Rakenduse turvalisuse haldamiseks kasutatakse tuntud ning pikalt arendatud raamistikku Spring Security. See võimaldab küllaltki lihtsasti ja mugavalt üles ehitada autentimise ja autoriseerimise süsteemi, millega saab kontrollida, milliste rollidega on võimalik teha serveripoolse rakenduse vastu päringuid. Võimalik on seadistada Spring Security raamistiku selliselt, et see kaitseks tuntumate rünnakute vastu, näiteks jõurünne (ingl. k. *brute-force attack*) ja murdskriptimine (ingl. k. *cross-site scripting*, lühendina XSS) [51].

Spring Security põhineb veebiteenuste filtritel, millest lähevad veebiteenuse päringud kindlas järjekorras läbi enne kontrollineri jõudmist. Selleks, et jõuda kontrollineri, tuleb edukalt läbida kõik filtrid. Filtreid on nii Spring Security raamistiku poolt kui ka võimalik ise juurde kirjutada, üle kirjutades mõne olemasoleva filtri meetodeid.

HTTP on seansita protokoll ehk ükski päring ei ole sõltuv eelnevast ega järgnevast [52]. Seetõttu tuleb päringutega kaasa panna autentimise ja autoriseerimise informatsioon. Selleks on olemas populaarne ja turvaline standard JSON Web Token (edaspidi JWT). JWT on standard, mis võimaldab kompaktsel viisil turvaliselt edastada klient- ja serverrakenduse vahel infot [53]. Esmalt saadetakse autentimise päring kliendi poolt serverile, genereeritakse vastav identiteedi token ning kliendi poolt pannakse igal järgneva päringuga see päises kaasa (Joonis 15).



Joonis 15. JWT tokeniga autoriseerimise ja autentimise voog.

JWT koosneb kolmest osast: päis (ingl. k. *header*), tookeni info (ingl. k. *payload*) ning allkiri (ingl. k. *signature*) ja need on eraldatud üksteisest punktiga. Päisesse kirjutatakse millist tüüpi tookeniga on tegu (alati JWT) ning millise algoritmiga on see allkirjastatud. Tookeni infosse on kirjutatud üldine kasutaja kohta käiv info, näiteks kasutajanimi, rollid, väljaandja ning aegumistähtaeg. Dekodeerida ilma algoritmi võtmesõnata on võimalik vaid päist ning informatsiooni osa. Viimane osa ehk allkiri verifitseeritakse serveri pool sama allkirjaga, et valideerida saadud sõnumi terviklikkus [53]. Lõputöös kasutatakse tookeni loomise ja valideerimise jaoks HS256 algoritmi ning aegumisajaks on määratud kolm tundi.

5.4 Klientrakenduse arendus

Järgnevates alapeatükkides antakse ülevaade klientrakenduse loomise, komponentide arendamise ning tekkinud probleemide ja leitud lahenduste kohta.

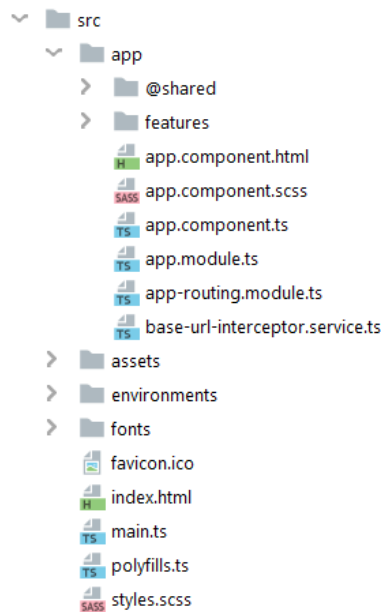
5.4.1 Klientrakenduse loomine

Klientrakendus loodi üheleherakendusena kaustades eelnevalt valitud Angulari raamistikku. Üheleherakendus tähendab seda, et veebirakendus laetakse ühe korra terviklikuna alla ning edaspidised nagiveerimised lehel on sellevõrra kiiremad [54]. Klientrakenduse põhi loodi kasutades Angular CLI-d (*Command Line Interface*), mis võimaldab mugaval viisil uut projekti luua.

Angulari üheks heaks omaduseks on moodulite olemasolu. Moodulid võimaldavad koguda seotud funktsionaalsused üheks kokku. Seotud funktsionaalsused võivad olla näiteks komponendid või teenused. Samuti on võimalik moodulite vahel jagada ka väiksemaid moduleid, näiteks rakenduse üleste komponentide jaoks loodud moodulit. Moodulid aitavad hoida rakenduse koodibaasi hästi struktureerituna, sest iga mooduliga seotud loogika ja komponendid on teistest eraldatud. Moodulid võimaldavad ka kiirendada rakenduse esmast laadimist *lazy loading*'u abil. See tähendab seda, et vajalikud moodulid laetakse kohe, kuid ülejäänud laetakse *on-demand* ehk siis kui alles vaja [55].

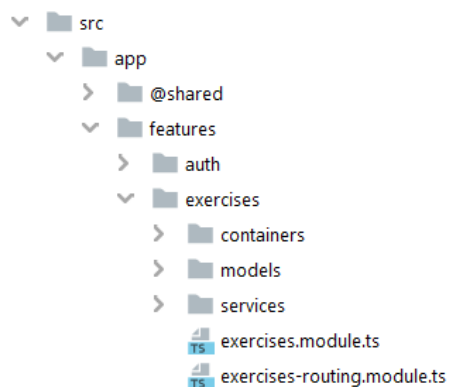
Seetõttu on ka loodud rakendus jaotatud ära mooduliteks, mis määrab ära osaliselt kaustade struktuuri (Joonis 16). *@Shared* kaust moodustab rakenduse jagatud teenuste

ja komponentide mooduli ning *features* kaust sisaldab erinevate komponentide mooduleid ehk *feature modules*.



Joonis 16. Klientrakenduse kaustade struktuur.

Features kaustas on igale komponendi kogumile loodud eraldi moodul koos marsruutimisega (ingl. k. *routing*). Joonis 17 näitab ühe mooduli struktuuri. Igal moodulil on kolm kausta: *containers*, *models* ja *services*. *Containers* ehk konteinerite kausta kuuluvad komponendid, mis tegelevad teenuste suhtluse, marsruutimise või oleku haldamisega. Iga komponendi juures on üldjuhul kolm faili: TypeScripti fail koodi ja loogika, HTML komponendi malli ning SCSS (*Sassy Cascading Style Sheet*) fail komponendi stiili jaoks. *Models* kaust on komponendis kasutatavate TypeScript objektide kirjeldamiseks. *Services* kaustas on teenused, mille läbi klientrakendus saab suhelda serveripoolse rakendusega.



Joonis 17. Ühe mooduli kaustade struktuur.

Selline kaustade struktuur võimaldab hoida komponentidega seotud loogika ühes kohas koos ning samuti parandada rakenduse laadimiskiirust jagades seotud vaated eraldi mooduliteks.

5.4.2 Suhtlus serveripoolse rakendusega

Veebirakenduse poolel kasutatakse serveripoolse rakendusega suhtlemiseks Angulari raamistikuga kaasa tulevat *HttpClient* teenust, mis võimaldab päringuid teha ning tulemust vastu võtta klientrakenduse poolel. Antud rakenduses on kõik päringud vaid sisse logitud kasutajatele (välja arvatud esmane autentimine), seega tuleb iga päringuga kaasa panna ka päises JWT token, mida valideeritakse serveri pool. Token saadetakse sisselogimise või registreerimise päringuga serveri poole ning salvestatakse brauseri sessioonihoidlasse (ingl. k. *session storage*).

Selle asemel, et panna token käsitsi iga päringuga kaasa, on võimalik luua Angular CLI tööriista kasutades päringute pealtkuulaja (ingl. k. *interceptor*), mis lisab selle iga päringuga kaasa (Joonis 18). Pealtkuulaja võtab autentimise teenuse abil hoidlast tokeni lisab sellele eesliidese “*Bearer*” ning paneb selle iga päringu päisesse *Authorization* võtme alla.

```
@Injectable()
export class AuthTokenInterceptor implements HttpInterceptor {
  constructor(private authService: AuthService) {}

  intercept(request: HttpRequest<unknown>, next: HttpHandler):
    Observable<HttpEvent<unknown>> {
    if (this.authService.isLoggedIn()) {
      request = request.clone({
        setHeaders: {
          Authorization: `Bearer ${this.authService.getToken()}`,
        },
      });
    }
    return next.handle(request);
  }
}
```

Joonis 18. Koodinäide autentimise pealtkuulajast.

Sissetulevad ja väljuvad päringud on alati samad ning neid ei saa muuta. Selleks, et lisada juurde päisesse andmeid, tuleb sellest luua kloon ning see edastada serveripoolsele rakendusele [56].

5.4.3 Kasutajate ligipääs rakendusele

Rakenduse funktsionaalsuste ligipääs on vaid autenditud kasutajatele ning lisaks veel on mõnele lehele ligipääs vaid kindla rolliga kasutajatel, näiteks ülesandeid ja tudengite õppimisaktiivsuse statistikat peaks nägema vaid õppejõu rolliga kasutaja. Selle lahendamiseks on olemas Angulari raamistikus valvurliidesed (ingl. k. *guards*), mis kontrollivad enne lehe laadimist, kas kasutajal on õigus lehele navigeerida.

Lehtedel võivad olla ka elemendid, mis on ainult teatud rollidele nähtavad. Selle asemel, et luua täiesti erinevaid komponente mõne elemendi tõttu, on kasutusele võetud Angulari direktiivid (ingl. k. *directive*). Tegemist on kohandatud HTML atribuutidega, mis saavad muuta DOM elementide välimust või käitumist [57]. Antud juhul loodi direktiiv, mis kontrollib, kas element peaks olema nähtav kasutajale rolli põhised.

5.4.4 Kasutajaliidese tõlked

Rakendus on esialgselt vaid eesti keeles, sest rakendus on loodud toetama eesti keeles läbiviidavat kursust. Sellest olenemata on rakendusele seatud funktsionaalne nõue, et koodibaasi keel peab olema inglisekeelne. Lisaks ei tohiks tekkida olukorda, kus uue keele lisamiseks on vaja teha muudatusi terves rakenduses.

Probleemi lahendamiseks võeti kasutusele Angulari raamistikule mõeldud internatsionaliseerimise teek *ngx-translate*. Teek võimaldab lihtsal viisil lisada rakendusele tõlkeid. Ülesseadmiseks tuleb importida teek rakenduse juurmoodulisse, sätestada vaikimisi keel ning määrata tõlgete laadimise koht [58]. Antud rakenduses keskendutakse esialgu vaid kasutajaliidese tõlgetele, seetõttu puudub hetkel vajadus serveripoolsetele tõlgetele.

Tõlkeid otsustati hoida klientrakenduse poolel JSON failis võti-väärtus paaridena. Nii on võimalik luua uue keele lisamiseks vaid uus tõlgetega fail ning teostada kasutajaliidese keele muutmise funktsionaalsus.

5.4.5 Mathlive teek

Mathlive teek sai valitud kui arvutustehete sisestamise tööriist. See võimaldab LaTeX formaadis kirjutada ning kuvada avaldise kasutajale. Kuigi teegi dokumentatsioon oli põhjalik ning näited lihtsad, oli selle kasutamine keerukam kui ette oli näha.

Nagu peatükis 4.2 mainitud, ei ole võimalik arvutiklaviatuurilt matrikseid sisestusväljale lisada ning tuli luua kohandatud virtuaalne klaviatuur, kus on olemas erinevate mõõtmetega tühjad matriksid. Autor eeldas, et võimalik on kasutada kohatäitja sümboleid, millele kirjutades asendub see kirjutatava sümboliga. Selgus, et kohatäitjat saab kasutada matriksites vaid teatud juhtudel: ühes reas võib maksimaalselt olla üks kohatäitja. Vastasel juhul väärtustub kogu matriksi sisu täpselt ühe kohatäitjaga ning see ei ole korrektne. Probleemi lahendamiseks otsustas autor võtta kasutusele kohatäitja sümbolina ringid, mida märgitakse TeX keeles kui „`\circ`“. Lahenduse eeliseks on see, et kasutajale kuvatakse selgelt kohta, kuhu tuleb väärtus kirjutada, kuid on ka puudusi. Lahendus mõjutab rakenduse kasutajakogemust, sest kirjutamiseks tuleb sümbol kustutada ning alles seejärel saab soovitud vastuse kirjutada. Autor leiab, et antud olukorras on see parim võimalik lahendus.

6 Tulemused

Järgnevates alapeatükkides kirjeldatakse antud lõputöö tulemust. Esmalt räägitakse lõputöö raames valminud funktsionaalsustest, seejärel testimise tulemustest ja võimalikest parandustest, mis tagasiside põhjal tehakse. Viimasena antakse soovitusi rakenduse edasiarenduse jaoks.

6.1 Loodud funktsionaalsused

Rakenduse esimeses versioonis on võimalik lahendada lineaaralgebra ülesandeid maatriksite elementaartehte kohta. Ülesannete lahendamisel on võimalik näha abistavat materjali, mis toetab teema paremat omandamist ning saada tagasisidet tehtud vigade kohta. Sellega täideti rakenduse põhieesmärk: luua harjutusväljak lineaaralgebra ülesannetele koos teoreetilise toega.

Samuti on võimalik luua hinnatavaid teste, mida kasutajad saavad lahendada ning mille järgi on õppejõul võimalik ka kursuse raames tudengitele hindeid anda. Kuigi tudengitel ei ole võimalik näha peale sooritamist enda sisestatud lahenduskäike ega selle õigsust, on selline võimalus olemas õppejõul.

Kõikidel rakenduse kasutajatel on võimalik näha nii üldist kui ka ülesandetüübist lähtuvat statistikat kasutusaktiivsuse kohta. Õppejõul on võimalik näha ka ülesannete õigesti-valesti lahendatud arvu, mis aitab saada ülevaadet, kui hästi on tudengid teema omandanud.

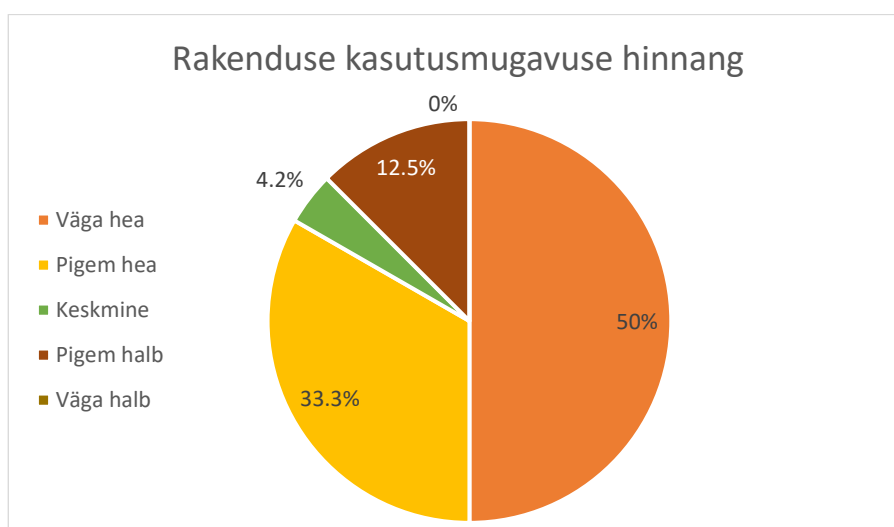
Kasutajate haldus oli antud lõputöös väiksema olulisusega. Esimeses versioonis on võimalik administraatori ning õppejõu rollidega kasutajatel muuta ka teiste kasutajate rolle: neid kasutajatele lisada ja neilt ka ära võtta. Igal kasutajal on võimalik muuta oma üldandmeid ning kas kasutajanime kuvatakse statistika lehel või ei. Kasutaja saab muuta ka parooli ja kustutada ka kasutajakonto.

Kuvatõmmised valminud rakendusest on leitavad lisades (Lisa 3). Rakendus on leitav aadressil <https://kallik.itcollege.ee/>.

6.2 Kasutajate tagasiside

Kokku testis rakendust 24 õppurit, neist 4 sessioonõppe ning 20 päevaõppe tudengit. Tudengitel paluti lahendada ülesandeid nii õigesti kui ka valesti, seejärel vastata *Google Forms* keskkonnas loodud küsitlusele (Lisa 4). Küsimustikus küsiti hinnangut 5 punkti skaalal rakenduse kasutusmugavusele, üldisele visuaalsele pildile ning ülesannete lahendamise kogemusele. Iga küsimuse juures oli võimalik põhjendada hinnangut. Lisaks küsiti kasutajatelt, mis neile rakenduses meeldis või ei meeldinud, ning koguti mõtteid edasiarenduse jaoks.

Peaaegu pooled vastajatest töid välja, et rakenduses on maatriksi elemente ebamugav sisestada, kuna eelnevalt tuleb kohatäitja ringisümbol ära kustutada ning alles seejärel on võimalik reaalne tulemus kirjutada. Nimetatud probleemi lahendamiseks antud hetkel head varianti ei ole ning sellise lahenduse kasutamine on põhjendatud peatükis 5.4.5. Sellest olenemata leidis 50% vastajatest, et rakendust on väga hea ning 33% leidis, et pigem hea kasutada (Joonis 19). Väga halvaks kasutusmugavust üksi vastaja ei hinnanud.



Joonis 19. Hinnangute jaotus rakenduse kasutusmugavusele.

Küsitluses toodi välja, et arvutiklaviatuurilt miinusmärki sisestades, lisati maatriksisse hoopis murd. Mathlive veebikomponent võimaldab teha sümbolite abil otseteid, näiteks kirjutades „/“ kuvatakse sisestusväljale tühi murd. Autor proovis samuti sisestada miinusmärki ning murru asemel tekkis sisestusvälja „-“. Autor arvab, et tagasisides välja toodud probleem võis tekkida ingliskeelse klaviatuuri kasutamisest, kus tavapärasel „-“

märgi kohal on hoopis „/“ ning seetõttu ka murd tehtesse sisestati. Kasutajatel, kellel selline probleem tekkis oli võimalik siiski virtuaalselt klaviatuurilt valida miinusmärki ning ülesanne seetõttu lahendamata ei jäänud.

Kriitilistest vigadest toodi välja, et mobiilirakendusel on osa vihjete ja valemite tekstist ekraanist väljaspool ning numbreid ei olnud võimalik sisestada, sest nutiseadme klaviatuur ei avanenud. Numbrite sisestamise probleemi lahendamiseks lisas autor virtuaalsele klaviatuurile ka numbrid, mida kohatäitjatele sisestada. Peale selle lisati valemi ja vihje ikoonile tekst, et kasutaja saaks selle funktsionaalsusest paremini aru.

Olenemata mõningatest parandussoovitustest leidsid kasutajad, et rakendus on lihtsa ning meeldiva disainiga (esmamulje keskmine hinnang on 4.5 punkti 5-st) ning navigeerimine oli lihtne ning vajalikud tegevused leiti kerge vaevaga ülesse. Samuti oli kasutajate arvates lahenduskäigu kontrollimisel pakutavate lahenduskäikude vigade tuvastamisest lahendamisel palju abi (19 vastajat leidis, et tagasiside on piisav, 4 leidis, et tagasiside võiks olla täpsem, 1 vastaja antud küsimusele ei vastanud). Samas toodi välja, et algajal võib rakenduses ülesannete lahendamisel probleeme tekkida arusaamisega ehk rakenduse kasutamise eelduseks peaks olema loengumaterjalidega tutvumine.

6.3 Edasiarenduse võimalused

Loodud lahendusele on mitmeid edasiarenduse võimalusi. Ülesannete lahendamise vaatest tuleks kindlasti võimaldada ka teiste kursuse raames lahendatavate lineaaralgebra ülesannete lahendamist. Samas oleks hea võimaldada ka ülejäänud kahe mooduli raames, et tekiks terviklik kursust toetav keskkond. Samuti toodi tudengite tagasisides välja, et rakenduses võiks olla eraldi sektsioon teoreetilise materjali jaoks.

Et motiveerida rohkem tudengeid antud rakenduses ülesandeid lahendama, võiks tekkida ka võistlusmoment, kus kasutajad saavad oma teadmised proovile panna. Näiteks ajalise piiranguga harjutustest, kus õigete vastuste eest antakse punkte juurde ja valede eest võetakse maha. Lõpuks moodustuks edetabel tudengitest, kes on sama ajaga rohkem või vähem punkte saanud.

Hetkel on lahendatud kasutajate autentimine eraldiseisvana Moodlest ning rakendusesiseselt. Et rakendus on mõeldud toetama kursust Kõrgem matemaatikat ning seal õpitut, võiks tulevikus toimuda rakenduse kasutajate autentimine Moodle keskkonna kaudu.

7 Kokkuvõte

Käesoleva lõputöö eesmärk oli luua interaktiivne abivahend, mis võimaldab tudengitel harjutada kursuse Kõrgem matemaatika raames ülesannete lahendamist, saada lahendamiseks vajalikku tagasisidet ning abi teoreetilise toe näol. Lisaks oli töö eesmärk pakkuda õppejõule hinnatavate testide korraldamise võimalust ja ülevaadet tudengite õppimisaktiivsusest. Olemasolevate lahenduste hulgast on raske leida kvaliteetset materjali, mis sobiks täielikult kokku kursuse raames käsitletavate teemadega ning pakuks interaktiivset, teoreetilise toega ülesannete harjutusväljakut.

Lõputöö eesmärgi saavutamiseks analüüsiti kursuse Kõrgem matemaatika ülesehitust, piirati lõputöö skoop ja uuriti interaktiivse õppekeskkonna loomise ning interaktiivse teoreetilise toe pakkumise võimalusi. Rakenduse arendusel kasutatavaks interaktiivse õppekeskkonna loomise meetodiks valiti *Action-Object-Input*, mille on loonud Marina Lepp koostöös kolleegidega. Loodavale lahendusele seati funktsionaalsed ja mittefunktsionaalsed nõuded ja lahenduse teostamiseks võrreldi erinevaid andmebaasi, serveripoolse- ja klientrakenduse tehnoloogiaid. Lõputöö käigus valminud rakenduse arendamiseks kasutati kliendipoolse rakenduse tehnoloogiatega TypeScripti ja Angulari ning serveripoolse rakenduse jaoks Javat ning Spring raamistikku.

Lahenduse valmimisel testiti selle kasutegurit käesoleval semestril ainekursusel osalevate tudengitega. Tudengid leidsid, et rakendus on lihtsa ning meeldiva disainiga ning sellisest rakendusest on kasu õppimisel. Laekunud tagasiside põhjal tehti ka mõningaid parandusi, et rakenduse kasutamine oleks edaspidi veelgi mugavam. Lõplik lahendus täitis seatud eesmärgid.

Antud rakenduse põhilise edasiarendusena tuleks lisada ka teiste kursuse raames õpetatavate teemade kohta harjutusülesannete lahendamisevõimalus koos vastava teoreetilise toega. Et motiveerida tudengeid antud rakenduses ülesandeid lahendama, võiks tekkida ka võistlusmoment, kus kasutajad saavad oma teadmised proovile panna.

Kasutatud kirjandus

- [1] Tallinna Tehnikaülikool, „e-Tugi juhised,“ [Võrgumaterjal]. Available: <https://moodle.taltech.ee/mod/book/tool/print/index.php?id=125831#ch3382>. [Kasutatud 27 veebruar 2022].
- [2] „Interaktiivne ülesannete lahendamise keskkond T-algebra,“ Tartu Ülikooli Arvutiteaduste instituut, Tartu, 2008.
- [3] Khan Academy, „About,“ [Võrgumaterjal]. Available: <https://www.khanacademy.org/about>. [Kasutatud 13 veebruar 2022].
- [4] Khan Academy, „Math: Linear algebra,“ [Võrgumaterjal]. Available: <https://www.khanacademy.org/math/linear-algebra>. [Kasutatud 13 veebruar 2022].
- [5] Udacity, „About Us,“ [Võrgumaterjal]. Available: <https://www.udacity.com/us>. [Kasutatud 15 veebruar 2022].
- [6] Udacity, „Linear Algebra Refresher Course,“ [Võrgumaterjal]. Available: <https://www.udacity.com/course/linear-algebra-refresher-course--ud953>. [Kasutatud 15 veebruar 2022].
- [7] P. Bellinga ja T. Gillebaart, „Our Vision,“ [Võrgumaterjal]. Available: <https://www.grasple.com/open/vision>. [Kasutatud 15 veebruar 2022].
- [8] Grasple, „Linear Algebra Exercises,“ [Võrgumaterjal]. Available: <https://www.grasple.com/math/linear-algebra>. [Kasutatud 15 veebruar 2022].
- [9] Symbolab, „What we do,“ [Võrgumaterjal]. Available: <https://www.symbolab.com/about>. [Kasutatud 15 veebruar 2022].
- [10] IXL Learning, „Our mission,“ [Võrgumaterjal]. Available: <https://www.ixl.com/company/mission>. [Kasutatud 15 veebruar 2022].
- [11] IXL Learning, „Algebra,“ [Võrgumaterjal]. Available: <https://www.ixl.com/math/algebra?>. [Kasutatud 15 veebruar 2022].
- [12] Coursera, „World-Class Learning for Anyone, Anywhere,“ [Võrgumaterjal]. Available: <https://about.coursera.org/how-coursera-works/>. [Kasutatud 16 veebruar 2022].
- [13] P. Gorbachenko, „What are Functional and Non-Functional Requirements and How to Document These,“ enkonix, 9 aprill 2021. [Võrgumaterjal]. Available: <https://enkonix.com/blog/functional-requirements-vs-non-functional/>. [Kasutatud 24 veebruar 2022].
- [14] M. Martin, „What is Non-Functional Requirement in Software Engineering? Types and Examples,“ Guru99, 18 veebruar 2022. [Võrgumaterjal]. Available: <https://www.guru99.com/non-functional-requirement-type-example.html>. [Kasutatud 24 veebruar 2022].
- [15] M. A. Ronan, „linear algebra,“ Britannica, 9 juuni 2017. [Võrgumaterjal]. Available: <https://www.britannica.com/science/linear-algebra>. [Kasutatud 24 veebruar 2022].
- [16] P. J. Olver ja C. Shakiban, Applied Linear Algebra, kd. 2, Springer International

- Publishing, 2018.
- [17] M. Issakova, „Solving of linear equations, linear inequalities and systems of linear equations in interactive learning environment,“ Tartu Ülikooli Kirjastus, Tartu, 2007.
 - [18] D. Lepp, „Algebraliste teisenduste harjutamise programm "Polynom",“ Tartu, 2003.
 - [19] L. Razzaq ja N. T. Heffernan, „Hints: Is It Better to Give or Wait to Be Asked?,“ Springer, Berlin, Heidelberg, 2010.
 - [20] Techopedia, „Interactivity,“ Techopedia, 19 August 2011. [Võrgumaterjal]. Available: <https://www.techopedia.com/definition/14429/interactivity>. [Kasutatud 27 veebruar 2022].
 - [21] E. Corral Abad, M. J. D.-J. E. Gómez García, P. M. Moreno-Marcos ja C. and Castejón Sisamon, „Improving the learning of engineering students with interactive teaching applications,“ 2021.
 - [22] Oracle, „Dynamic typing vs. static typing,“ Oracle Corporation, märts 2015. [Võrgumaterjal]. Available: https://docs.oracle.com/cd/E57471_01/bigData.100/extensions_bdd/src/cext_transform_typing.html. [Kasutatud 10 märts 2022].
 - [23] M. Bhatnagar, „Magic lies here - Statically vs Dynamically Typed Languages,“ Medium, 9 September 2019. [Võrgumaterjal]. Available: <https://medium.com/android-news/magic-lies-here-statically-typed-vs-dynamically-typed-languages-d151c7f95e2b>. [Kasutatud 10 märts 2022].
 - [24] J. Hartman, „Java vs C# – 10 Key Differences between Java and C#,“ Guru99, 5 märts 2022. [Võrgumaterjal]. Available: <https://www.guru99.com/java-vs-c-sharp-key-difference.html>. [Kasutatud 16 märts 2022].
 - [25] „Spring Framework Overview,“ [Võrgumaterjal]. Available: <https://docs.spring.io/spring-framework/docs/current/reference/html/overview.html>. [Kasutatud 16 märts 2022].
 - [26] Microsoft, „ASP.NET,“ Microsoft, [Võrgumaterjal]. Available: <https://dotnet.microsoft.com/en-us/apps/aspnet>. [Kasutatud 16 märts 2022].
 - [27] G. Dreimanis ja O. Bolgurtseva, „Why You Should Choose TypeScript Over JavaScript,“ Serokell, 18 juuni 2020. [Võrgumaterjal]. Available: <https://serokell.io/blog/why-typescript>. [Kasutatud 16 märts 2022].
 - [28] B. Popper ja D. Gibson, „Most popular technologies: Web frameworks,“ Stack Overflow, 2 august 2021. [Võrgumaterjal]. Available: <https://insights.stackoverflow.com/survey/2021#most-popular-technologies-webframe>. [Kasutatud 10 märts 2022].
 - [29] D. Guo, „The history and legacy of jQuery,“ LogRocket, 13 august 2019. [Võrgumaterjal]. Available: <https://blog.logrocket.com/the-history-and-legacy-of-jquery/>. [Kasutatud 12 mai 2022].
 - [30] MathJax, „MathJax: Beautiful and accessible math in all browsers,“ MathJax, [Võrgumaterjal]. Available: <https://www.mathjax.org/>. [Kasutatud 17 märts 2022].
 - [31] CortexJs, „A Web Component for Math Input,“ 29 aprill 2020. [Võrgumaterjal]. Available: <https://cortexjs.io/mathlive/>. [Kasutatud 17 märts 2022].
 - [32] „KaTeX - The fastest math typesetting library for the web,“ [Võrgumaterjal]. Available: <https://katex.org/>. [Kasutatud 16 märts 2022].
 - [33] „Web Components,“ [Võrgumaterjal]. Available: <https://developer.mozilla.org/en->

- US/docs/Web/Web_Components. [Kasutatud 20 märts 2022].
- [34] Tartu Ülikool, „Veel valemite vormistamise võimalusi: LaTeX,“ [Võrgumaterjal]. Available: <https://sisu.ut.ee/valemid/tex>. [Kasutatud 27 märts 2022].
- [35] „Why Spring?,“ [Võrgumaterjal]. Available: <https://spring.io/why-spring>. [Kasutatud 18 märts 2022].
- [36] Tutorialspoint, „Hibernate - ORM Overview,“ [Võrgumaterjal]. Available: https://www.tutorialspoint.com/hibernate/orm_overview.htm. [Kasutatud 18 märts 2022].
- [37] M. Liang, „Understanding Object-Relational Mapping: Pros, Cons, and Types,“ altexsoft, 11 märts 2021. [Võrgumaterjal]. Available: <https://www.altexsoft.com/blog/object-relational-mapping/>. [Kasutatud 18 märts 2022].
- [38] Knowledge Factory, „Top 5 Java ORM tools - 2022,“ Knowledge Factory, 16 september 2021. [Võrgumaterjal]. Available: <https://www.knowledgefactory.net/2021/09/top-java-orm-tools-20XX.html>. [Kasutatud 7 mai 2022].
- [39] Geeks for Geeks, „Difference between Spring MVC and Spring Boot,“ Geeks for Geeks, 23 november 2020. [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/difference-between-spring-mvc-and-spring-boot/>. [Kasutatud 18 märts 2022].
- [40] „Colt,“ MVNrepository, [Võrgumaterjal]. Available: <https://mvnrepository.com/artifact/colt/colt>. [Kasutatud 18 märts 2022].
- [41] „Apache Commons Math,“ MVNRepository, [Võrgumaterjal]. Available: <https://mvnrepository.com/artifact/org.apache.commons/commons-math3>. [Kasutatud 18 märts 2022].
- [42] „Weka Stable,“ MVNrepository, [Võrgumaterjal]. Available: <https://mvnrepository.com/artifact/nz.ac.waikato.cms.weka/weka-stable>. [Kasutatud 18 märts 2022].
- [43] „Most Popular Technologies: Databases,“ Stack Overflow, 2021. [Võrgumaterjal]. Available: <https://insights.stackoverflow.com/survey/2021#most-popular-technologies-database>. [Kasutatud 18 märts 2022].
- [44] M. Drake ja ostezer, „SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems,“ Digital Ocean, 9 märts 2022. [Võrgumaterjal]. Available: <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>. [Kasutatud 18 märts 2022].
- [45] Figma, „The modern interface design tool,“ Figma, [Võrgumaterjal]. Available: <https://www.figma.com/ui-design-tool/>. [Kasutatud 7 mai 2022].
- [46] M. F. Mustafar ja M. A. Dzulkifli, „The Influence of Colour on Memory Performance: A Review,“ Department of Psychology, International Islamic University Malaysia, Selangor, 2013.
- [47] Vertabelo SA, „Web Based Data Modeler for Professionals,“ Vertabelo SA, [Võrgumaterjal]. Available: <https://vertabelo.com/>. [Kasutatud 7 mai 2022].
- [48] M. Richards, „Chapter 1. Layered Architecture,“ O'Reilly Media, Inc, 2015. [Võrgumaterjal]. Available: <https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html>. [Kasutatud 1 aprill 2022].

- [49] B. Krebs, „Database Versioning with Flyway and Java,“ auth0, 28 aprill 2017. [Võrgumaterjal]. Available: <https://auth0.com/blog/incrementally-changing-your-database-with-java-and-flyway/>. [Kasutatud 5 aprill 2022].
- [50] IBM Cloud Education, „REST APIs,“ IBM, 6 aprill 2021. [Võrgumaterjal]. Available: <https://www.ibm.com/cloud/learn/rest-apis>. [Kasutatud 16 aprill 2022].
- [51] Y. K. Erinç, „Spring Security Overview,“ auth0, 15 aprill 2021. [Võrgumaterjal]. Available: <https://auth0.com/blog/spring-security-overview/>. [Kasutatud 6 aprill 2022].
- [52] GeeksforGeeks, „Difference between Stateless and Stateful Protocol,“ GeeksforGeeks, 1 märts 2021. [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/difference-between-stateless-and-stateful-protocol/>. [Kasutatud 6 aprill 2022].
- [53] auth0, „Introduction to JSON Web Tokens,“ auth0, [Võrgumaterjal]. Available: <https://jwt.io/introduction>. [Kasutatud 6 aprill 2022].
- [54] G. Singhal, „Why Do We Need Single-page Applications?,“ Pluralsight, 9 aprill 2022. [Võrgumaterjal]. Available: <https://www.pluralsight.com/guides/why-do-we-need-a-single-page-application>. [Kasutatud 16 aprill 2022].
- [55] D. Saha, „Feature Modules in Angular – Why Required and When to Use?,“ DZone, 1 detsember 2020. [Võrgumaterjal]. Available: <https://dzone.com/articles/feature-modules-in-angular-why-required-and-when-t>. [Kasutatud 16 aprill 2022].
- [56] K. Schuchard, „3 ways to use Angular HTTP Interceptors,“ Briebug, [Võrgumaterjal]. Available: <https://blog.briebug.com/blog/3-ways-to-use-angular-http-interceptor>. [Kasutatud 16 aprill 2022].
- [57] Google LLC, „Attribute directives,“ Google LLC, 28 veebruar 2022. [Võrgumaterjal]. Available: <https://angular.io/guide/attribute-directives>. [Kasutatud 18 aprill 2022].
- [58] NGX-Translate, „@ngx-translate/core,“ 8 november 2021. [Võrgumaterjal]. Available: <https://github.com/ngx-translate/core>. [Kasutatud 21 aprill 2022].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Kärt Allik

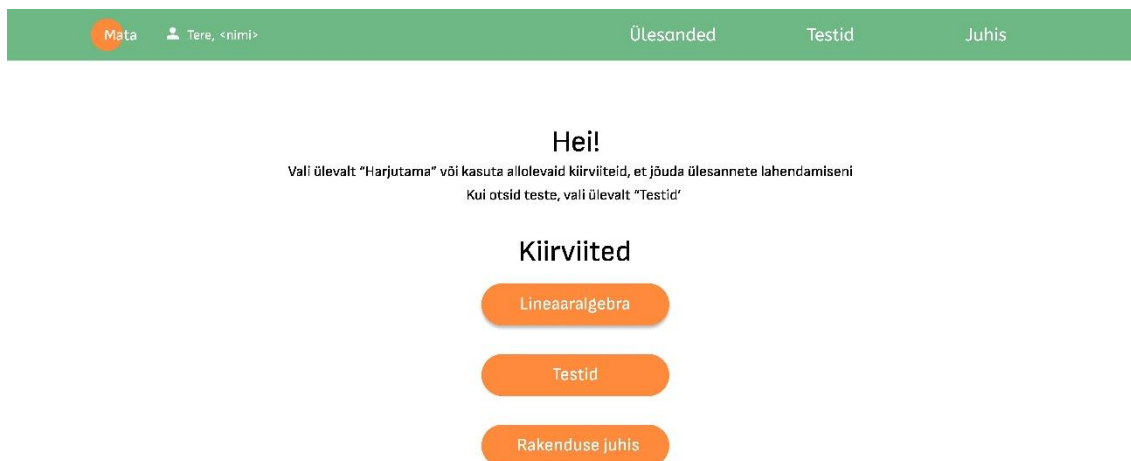
1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Veebirakendus kursuse Kõrgem matemaatika toetamiseks”, mille juhendaja on Kristiina Hakk
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse 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.

16.05.2022

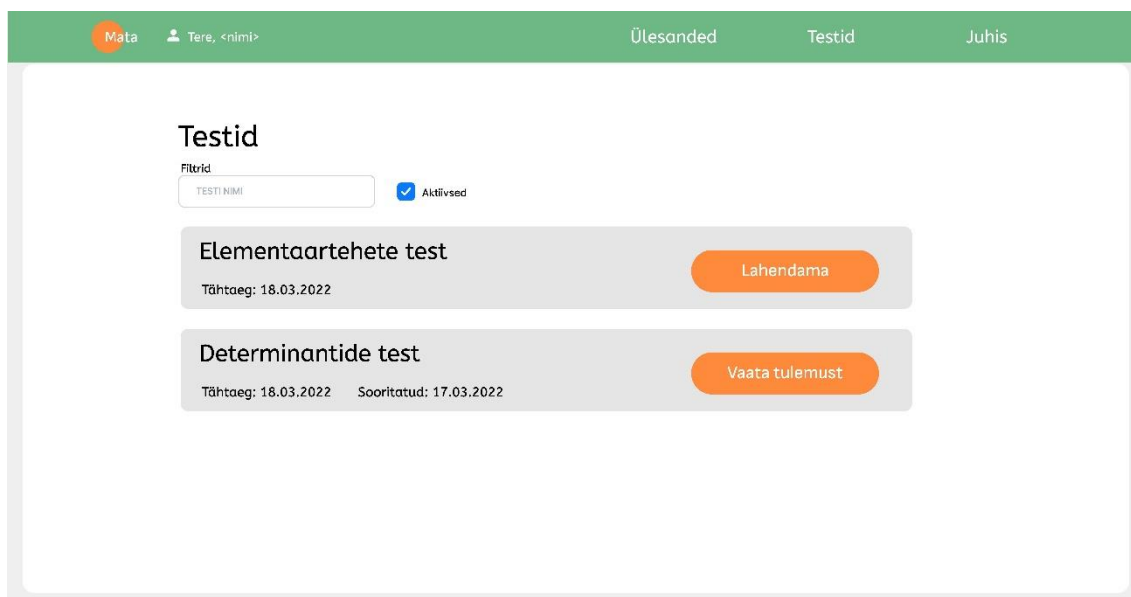
¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või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

Lisa 2 – Rakenduse vaadete prototüübid

Esileht



Testide koondvaade



Testi statistika üldine vaade

Mata Tere, <nimi>

Ülesanded Testid Juhis

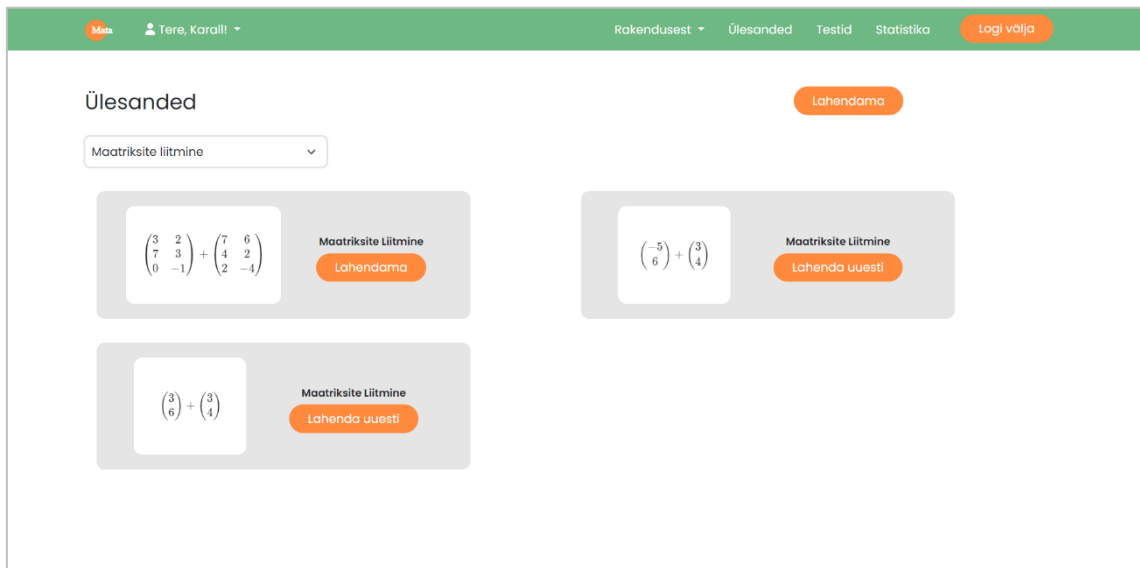
Statistika

Test: elementaartehete test

Tagasi testi juurde

Nimi	Kasutajanimi	Punktid	Tulemus	
Mark	Otto	10/20	50%	Vaata
Jacob	Thornton	5/20	25%	Vaata
Larry	the Bird	20/20	100%	Vaata

Lisa 3 – Valminud rakenduse kuvatõmmised



Tagasi

Ülesanne: Liida maatriksid

Avaldise vahetehtes on tehtud viga, maatriksi elemendi a_{21} vahetehe "6-4" ei ole õige!

$$\begin{pmatrix} -5 \\ 6 \end{pmatrix} + \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} -5+3 \\ 6-4 \end{pmatrix}$$

Maatriksi sisestamiseks vali avanevalt klaviatuuriit õigete mõõtmetega maatriksi põhi.

Valem \sqrt{x}

Vihje $?$

Kontrolli \checkmark

Vastus \checkmark

Lõpeta lahendamine ⌂



Tagasi

Ülesanne: Liida matriksid

Matriksite liitmisel on vastuses tehtud viga, matriksi elemendi a_{11} väärtus ei ole 7

$$\begin{pmatrix} 3 \\ 6 \end{pmatrix} + \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 3+3 \\ 4+6 \end{pmatrix} = \begin{pmatrix} 7 \\ 10 \end{pmatrix}$$

Matriksi sisestamiseks vali avanevalt klaviatuurilt õigete mõõtmetega matriksi põhi.

- Valem \sqrt{x}
- Vihje $?$
- Kontrolli \checkmark
- Vastus \checkmark
- Lõpeta lahendamine ✕

Tagasi

Ülesanne: Transponeeri matriks

$$\begin{pmatrix} -5 \\ 5 \end{pmatrix}^T =$$

Matriksi sisestamiseks mõõtmetega matriks

Transponeerimisel muutub matriksi dimensioon

- Valem \sqrt{x}
- Vihje $?$
- Kontrolli \checkmark
- Vastus \checkmark
- Lõpeta lahendamine ✕

Lisa 4 – Tagasiside küsimustik

Tagasiside rakendusele

Hei! Olen Kärt, IT süsteemide arenduse 3 kursuse tudeng ja tegin lõputöö raames kursusele Kõrgem Matemaatika veebirakenduse.

Rakenduse eesmärk on pakkuda lisa õppimis- ja harjutamisvõimalust kursuse kõrgem matemaatika raames.

Rakendus on leitav siit: <https://kallik.itcollege.ee>

Selleks, et hinnata rakenduse kasulikkust, palun Teil lahendada vähemalt ühte ülesannet ning seda teha nii "valesti" kui "õigesti", et näha, millist tagasisidet rakendus lahendamise käigus pakub.

1. Kas olete päeva- või sessioonõppes?

- Päevaõppes
- Sessioonõppes

2. Millise seadmega veebilehte kasutasite?

- Arvuti (sülearvuti, PC, muu suure ekraaniga seade)
- Tahvelarvuti (keskmise suurusega ekraan)
- Nutiseade (väike ekraan)

3. Milliseid ülesandeid lahendasite?

- Maatriksite liitmine
- Maatriksite lahutamine
- Maatriksite korrutamine
- Maatriksi korrutamine skalaariga
- Rea- ja veeruvektori korrutamine
- Maatriksi transponeerimine
- Ülesandeid ei lahendanud, aga tahtsin küsimustikule vastata.

4. Tähelepanekuid lahendamise kohta? (Kas tagasiside ei olnud piisav? Kui mugav oli ülesannet lahendada? Rakenduse pakutud vastus oli vale?)

5. Kas lahendamise vahetehte kontroll andis piisavat tagasisidet, et leida lahenduskäigus viga?

- 1 – ei andnud
- 5 – andis piisavat

6. Soovikorral täpsustage eelmist vastust

7. Rakenduse üldine visuaalne pilt (esmamulje) oli:

- 1 – väga halb
- 5 – väga hea

8. Soovi korral täpsustage eelmist vastust

9. Kuidas hindate rakenduse kasutusmugavust?

- 1 – väga halb
- 5 – väga hea

10. Soovi korral täpsustage eelmist vastust

11. Rakenduses oli lihtne navigeerida ning leida vajalikke kohti ülesse

- 1 – väga raske
- 5 – väga lihtne

12. Soovi korral täpsustage eelmist vastust

13. Kasutajaliidese nupud-lahtrid-lingid olid arusaadava eesmärgiga.

- 1 – ei olnud
- 5 – olid

14. Soovi korral täpsustage eelmist vastust.

15. Mida tahaksite veel selles rakenduses näha? (Mis oleks tulevikus tudengitele kasulik).

16. Kas oli midagi, mis rakenduse puhul väga meeldis?

17. Kas oli midagi, mis rakenduse puhul ei meeldinud üldse?

18. Kas soovitaksite rakendust teistele tudengitele?

- 1 – Ei soovitaks kindlasti
- 5 – Kindlasti soovitaksin

19. Soovi korral täpsustage eelmist vastust

20. Muu positiivne-neutraalne-negatiivne tagasiside rakenduse kohta

21. Kas rakenduse kasutamisel tekkis vigu? Milliseid? Kuidas?