

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Kristjan Junus 164011IAPB

**GÜMNASISTIDE MATEMAATIKA  
ÜLESANNETE ISESEISVA LAHENDAMISE  
KESKKONNA KASUTAJALIIDESE  
ARENDAMINE**

Bakalaurusetöö

Juhendaja: Gunnar Piho  
PhD

Tallinn 2021

## **Autorideklaratsioon**

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

Autor: Kristjan Junus

17.05.2021

## Annotatsioon

Töö eesmärgiks on luua matemaatiliste ülesannete lahendamise keskkond, mis kuvab läbi Wordi sisestatud ülesandeid ja matemaatilisi valemeid. Wordis võib sisestada ükskõik milliseid matemaatilisi valemeid, tabeleid või jooniseid. Kasutajaliideses peab saama navigeerida erinevate kogumike ning ülesannete vahel. Ülesandeid saab lahendada valikvastuste abil. Kasutajaliides peab toetama erinevaid brausereid ning erinevaid ekraani suuruseid.

Taustauuringus selgus, et olemasolevad lahendused ei sobi kokku piisaval määral kliendi poolt seatud kriteeriumitega. Töö käigus tehti analüüs põhinõuete väljaselgitamiseks ning tehti vajalikud ettevalmistused arendustööks.

Töö tulemusena valmis klientrakendus, mida gümnasistid saavad iseseisvalt kasutada matemaatika eksamiks ettevalmistamisel. Rakendus on kättesaadaval aadressil <https://matemaatika.taltech.ee/>.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 28 leheküljel, 6 peatükki, 11 joonist, 8 tabelit.

# Abstract

## **Development of the User Interface for the Independent Solving of Mathematics Tasks for Upper Secondary School Students**

The purpose of this bachelor's thesis is to create an application for solving mathematical problems. The application would consist of collections of tasks. Every task would have answers and one or more solutions for the task. Tasks can be solved using multiple choice answers and upon choosing correct answer, related solutions would be shown. The whole tasks collection would be entered in the system using Word files, where the input would be processed into tasks, answers and solution entities. Word file can contain complicated formulas, tables or figures.

The background study revealed that the existing applications for solving a similar problem were not sufficiently compatible with the criteria set by the client. During the thesis, an analysis was made to find out the basic requirements and the necessary preparations for starting the development was made. During the development, used frameworks and tools, architectural choices and developed functionalities were discussed. After the development process, the result was validated against requirements set in the analysis.

As a result of this thesis a client application was completed, which upper secondary school students can use independently in preparing for the mathematics exam. The application uses client-server architecture. Front-end of the application was made using Angular and it communicates with back-end using API (*Application Programming Interface*). The application is available at <https://matemaatika.taltech.ee/>.

The thesis is in Estonian and contains 28 pages of text, 6 chapters, 11 figures, 8 tables.

## Lühendite ja mõistete sõnastik

LaTeX	<i>Lamport's TeX</i> , tehnilise ja teadusliku dokumentide jaoks kasutatav märgenduskeel.
IDE	<i>integrated development environment</i> , integreeritud programmeerimiskeskond
npm	<i>Node Package Manager</i> , Node paketi haldur
CSS	<i>Cascading Style Sheets</i> , kaskaadlaadistik
HTML	<i>HyperText Markup Language</i> , hüpertexti märgistuskeel
SCSS	<i>Sassy CSS</i> , stiili keel, mida kasutatakse CSS keele võimekuse suurendamiseks
API	<i>Application Programming Interface</i> , programmi liides – defineerib suhtluse erinevate rakenduste vahel
REST	<i>Representational state transfer</i> , tarkvararhitektuuri laad
HTTP	<i>Hypertext Transfer Protocol</i> , hüpertexti edastusprotokoll
JSON	<i>JavaScript Object Notation</i> , lihtsustatud andmevahetusvorming
PNG	<i>Portable Network Graphics</i> , rastergraafika failivorming
MathML	<i>Mathematical Markup Language</i> , matemaatikaline XML
XML	<i>Extensible Markup Language</i> , laiendatav märgistuskeel
OWASP	<i>Open Web Application Security Project</i> , tarkvara turvalisusele pühendunud organisatsioon
XSS	<i>Cross Site Scripting</i> , murdskriptimine
HTTPS	<i>Hyper Text Transfer Protocol Secure</i> , turvaline HTTP
CORS	<i>Cross-Origin Resource Sharing</i> , mehhanism, mis lubab brauseril pääseda ligi teise aadressiga domeeni ressurssidele
CDN	<i>Content Delivery Network</i> , sisuedastusvõrk
CNAME	<i>Canonical Name</i> , alias ümbersuunamiseks teisele lehele
TLS	<i>Transport Layer Security</i> , transpordikihi turbeprotokoll
PC	<i>Personal Computer</i> , personaalarvuti

# Sisukord

1	Sissejuhatus .....	10
1.1	Probleem.....	10
1.2	Eesmärk .....	11
1.3	Töö struktuur .....	11
2	Taustauuring .....	12
2.1	Kasutajaliidesed.....	12
2.1.1	Harno Moodle.....	12
2.1.2	E-koolikott.....	13
2.1.3	SOWISO.....	14
2.2	Word dokumendist ülesannete lugemine.....	14
2.3	Taustuuringu kokkuvõte .....	15
3	Analüüs.....	16
3.1	Objekti kirjeldus .....	16
3.2	Töö protsess.....	16
3.3	Üldine arhitektuur.....	17
3.4	Funktsionaalsed nõuded .....	17
3.5	Mittefunktsionaalsed nõuded.....	18
3.6	Näidismudel.....	18
4	Klientrakenduse arendamine .....	19
4.1	Koodi stiil .....	19
4.1.1	LIFT.....	19
4.2	Kasutatud tehnoloogiad .....	20
4.2.1	Angular .....	20
4.2.2	Tslint.....	21
4.2.3	SCSS ja Bootstrap .....	21
4.2.4	PrimeNG.....	22
4.2.5	Browserlist.....	22
4.2.6	Jasmine ja Karma.....	22
4.3	Projekti struktuur .....	23
4.4	Klient-server kommunikatsioon .....	25
4.4.1	RESTful API .....	25

4.5	Konfiguratsioon.....	26
4.6	Ülesannete kuvamine.....	27
4.7	Vaated.....	28
4.8	Turvalisus .....	32
4.8.1	HTTPS.....	32
4.8.2	CORS.....	33
4.9	Erinevad ekraanid .....	33
4.10	Rakenduse paigaldamine .....	33
5	Kasutajaliidese valideerimine.....	34
5.1	Brauserid ja erinevad seadmed .....	34
5.2	Üksustestid .....	35
5.3	Turvalisus .....	35
5.3.1	XSS.....	35
5.3.2	SQL süstimine .....	36
5.3.3	Komponendid teadaolevate haavatavustega.....	36
5.3.4	Ebapiisav logimine ja seire.....	36
6	Kokkuvõte .....	38
	Kasutatud kirjandus .....	39
	Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	41
	Lisa 2 – Näidismudeli avavaade.....	42
	Lisa 3 – Näidismudeli ülesande lahendamise vaade valikvastustega.....	43
	Lisa 4 – Näidismudeli lahenduskäigu vaade .....	44
	Lisa 5 – Näidismudeli mobiilivaates avavaade .....	45

## Jooniste loetelu

Joonis 1. Üldine arhitektuur.....	17
Joonis 2. Tuuma kausta struktuur. ....	23
Joonis 3. Jagatud kausta struktuur. ....	24
Joonis 4. Põhiloode kausta struktuur. ....	24
Joonis 5. Stiili kaust.....	25
Joonis 6. Lehekülgedeks jagamise mudel.....	26
Joonis 7. Kasutajaliidese avavaade.....	28
Joonis 8. Kogumiku ülesannete nimekiri. ....	29
Joonis 9. Kogumiku ülesande vaade koos valikvastustega .....	30
Joonis 10. Ülesande vaade koos lahenduskäikudega.....	31
Joonis 11. Ülesande detail vaade mobiilis.....	32



## Tabelite loetelu

Tabel 1. Portaali Harno Moodle positiivsed omadused.....	12
Tabel 2. Portaali Harno Moodle negatiivsed omadused.....	13
Tabel 3. Portaali E-koolikott positiivsed omadused.....	13
Tabel 4. Portaali E-koolikott negatiivsed omadused.....	13
Tabel 5. Portaali SOWISO positiivsed omadused.....	14
Tabel 6. Portaali SOWISO negatiivsed omadused.....	14
Tabel 7. LIFT põhimõtted .....	19
Tabel 8. Testimiseks kasutatud seadmed ning brauserid.....	34

# 1 Sissejuhatus

Iga aasta väljastatakse ettevõtte Kirjastus Argo poolt „Eksaminandile matemaatika riigieksamist“ ülesannete kogumik<sup>1</sup>. Kogumik sisaldab eksamite sooritamiseks vajalikke põhiteadmisi ja -oskuseid, mis on toetatud näiteülesannetega. Lisaks on iga teema kohta enesekontrolliks harjutusülesanded. Harjutusülesanne koosneb eraldi ülesandest, lahenduskäigust ning vastusest [1].

Ülesannete kogumike autor iga aasta on matemaatika õpetaja. Ülesannete tegemiseks kasutab õpetaja Wordi. Wordis on olemas õpetaja jaoks kõik vajalik, et sisestada keerulisemaid matemaatilisi valemeid, tabelleid ja jooniseid.

## 1.1 Probleem

Kogumike autori soov on, et oleks olemas keskkond, kuhu autor saab vähese vaevaga üles laadida oma koostatud Wordi faili. Failist lugedes koostatakse keskkonna poolt ülesanded koos valikvastustega ning lahenduskäikudega. Antud lõputöö raames arendatakse ülesannete lahendamise pool, kus õpilane saab enesekontrolliks lahendada ülesandeid valikvastuste abil. Kui keskkond osutub edukaks, ehk sellel on tulevikus kasutajad ning see suurendab kogumike müüki, siis tahab kirjastus keskkonna edasi arendamise võimalust, et lisada näiteks maksumüür (ingl *paywall*) või keerulisemat tagasiside ja soovitusfunktsionaalsust.

Lõputöö raames realiseeritakse kasutajaliides, kus peamised funktsionaalsused on kogumike ja ülesannete vahel navigeerimine, sobivate ülesannete valimine lahendamiseks, valikvastuste kuvamine ning õige vastuse korral lahendi kuvamine. Serverirakendus koos ülesannete sisestamisega arendatakse eraldiseisvalt antud tööst.

---

<sup>1</sup> [argokirjastus.ee/toode/eksaminandile-matemaatika-riigieksamist-2020/](http://argokirjastus.ee/toode/eksaminandile-matemaatika-riigieksamist-2020/)

## **1.2 Eesmärk**

Esimeseks eesmärgiks on luua kasutajaliides gümnaasistide matemaatika ülesannete iseseisvaks lahendamiseks. Rakendus on kasutatav arvuti ekraanidest kuni mobiilideni, toetades ka suurema osa kasutusel olevaid veebibrausereid.

Teiseks eesmärgiks oleks kuvada veebilehel ka keerulisemaid matemaatilisi valemeid, mida õpetaja on läbi Wordi sisestanud. Lisaks valemitele peavad olema kuvatud ka kõik tabelid ja joonised, mis on kogumikesse sisestatud. Teise eesmärgi saavutamiseks peab tegema koostööd serverirakenduse arendajaga, et võimalikult efektiivne lahendus saavutada.

## **1.3 Töö struktuur**

Lõputöö koosneb kuuest peatükist. Teises peatükis tutvustatakse olemasolevaid lahendusi ja võrreldakse nende pakutavaid võimalusi kliendi vajadustega. Kolmandas peatükis kirjeldatakse kogu kasutajaliidese arenduseks tehtud eeltöö. Neljandas peatükis kirjeldatakse kasutajaliidese arendust. Viiendas peatükis kirjeldatakse lõpptulemuse valideerimist. Kuuendas peatükis on töö kokkuvõte.

## 2 Taustauuring

Peatükis kirjeldatakse olemasolevaid lahendusi, analüüsitakse nende vastavust kliendi nõuetele, tuues välja positiivsed ja negatiivsed omadused. Peatükk on jaotatud kahte ossa. Esimene peatükk kirjeldab sarnast otstarbekust toetavaid kasutajaliideseid ja teine osa Wordi dokumendist matemaatiliste ülesannete väljalugemist võimaldavad rakendused.

### 2.1 Kasutajaliideseid

Kasutajaliideseid analüüsidest tuuakse välja selle positiivsed ja negatiivsed aspektid, et hiljem oleks võimalik neid antud töö raames arendades kasutada. Portaale vaadatakse nii isiku seisukohast, kes peab looma matemaatika kogumikke, kui ka koole, kes peaksid lõpptoote kasutusele võtma.

#### 2.1.1 Harno Moodle

Harno Moodle<sup>1</sup> on Eestis kättesaadav veebipõhise õppe keskkond, mis on mõeldud kasutamiseks kõikidele õppeainetele. Järgnevas tabelis on esitatud portaali positiivsed omadused (Tabel 1).

Tabel 1. Portaali Harno Moodle positiivsed omadused.

Positiivne omadus	Põhjendus
Modernne mulje	Veebileht on modernse välimusega, vajalik informatsioon on lihtsasti leitav
Lai kasutusväli	Kool saab võtta kasutusele ainult ühe platvormi, kus on koos kõik ained, mis lihtsustab rakenduse kasutuselevõttu

---

<sup>1</sup> moodle.edu.ee

Järgmises tabelis on välja toodud negatiivsed omadused (Tabel 2).

Tabel 2. Portaali Harno Moodle negatiivsed omadused.

Negatiivne omadus	Põhjendus
LaTeX (ingl <i>Lamport's TeX</i> )	Matemaatika ülesannete sisestamisel on kasutusel LaTeX, mis teeb ülesannete sisestamise õpetaja jaoks keerulisemaks
Sihtgrupp on koolid	Tasuta saavad keskkonda kasutada koolid, eraettevõtted või -isikud peavad maksma kasutusõiguste eest
Lisaarenduse võimalus puudub	Vastavalt kliendi nõudele, puudub võimalus täiendada veebilehte vastavalt vajadustele

### 2.1.2 E-koolikott

E-koolikott<sup>1</sup> on Eestis kättesaadav õppematerjalide portaal. Funktsionaalsuselt sarnane Harno Moodlele. Järgnevas tabelis on esitatud portaali positiivsed omadused (Tabel 3).

Tabel 3. Portaali E-koolikott positiivsed omadused.

Positiivne omadus	Põhjendus
Modernne mulje	Autori hinnangul kõige modernsema välimusega veebileht Eestis, sisuhaldus on lihtne aga samas põhjalik
Lai kasutusväli	Kool saab võtta kasutusele ainult ühe platvormi, kus on koos kõik ained, mis lihtsustab rakenduse kasutuselevõttu

Järgmises tabelis on välja toodud negatiivsed omadused (Tabel 4).

Tabel 4. Portaali E-koolikott negatiivsed omadused.

Negatiivne omadus	Põhjendus
LaTeX (ingl <i>Lamport's TeX</i> )	Matemaatika ülesannete sisestamisel on kasutusel LaTeX, mis teeb ülesannete sisestamise õpetaja jaoks keerulisemaks
Lisaarenduse võimalus puudub	Vastavalt kliendi nõudele, puudub võimalus täiendada veebilehte vastavalt vajadustele

---

<sup>1</sup> e-koolikott.ee

### 2.1.3 SOWISO

SOWISO<sup>1</sup> on Hollandis tegutsev ettevõtte, veebileht on inglise keeles kättesaadav. SOWISO on spetsialiseerunud matemaatika ülesannete lahendamise keskkonnaks. Järgnevas tabelis on esitatud portaali positiivsed omadused (Tabel 5).

Tabel 5. Portaali SOWISO positiivsed omadused.

Positiivne omadus	Põhjendus
Modernne mulje	Veebileht on modernse välimusega, vajalik informatsioon on lihtsasti leitav
Ainult matemaatikale mõeldud	Veebileht toetab palju erinevaid matemaatika ülesannete variante
Vastuste analüüsimine	Õpilaste vastuseid analüüsitakse ning antakse õpetajale tagasisidet näiteks nõrkuste kohta

Järgmises tabelis on välja toodud negatiivsed omadused (Tabel 6).

Tabel 6. Portaali SOWISO negatiivsed omadused.

Negatiivne omadus	Põhjendus
Ülesannete sisestamine veebilehes	Ülesannete sisestamiseks peab kasutama veebilehel asuvat integreeritud klaviatuuri
Lisaarenduse võimalus puudub	Vastavalt kliendi nõudele, puudub võimalus täiendada veebilehte vastavalt vajadustele
Sisse logimine kohustuslik	Uute kasutajate kaasamine on raskendatud

## 2.2 Word dokumendist ülesannete lugemine

Vajadus on Word dokumenti lugeda ning teisendada seda sisuks, mida saab klientrakenduse pool kuvada. Kriteeriumid on, et lahendus toetaks matemaatiliste valemite lugemist ning, et õpetaja töövoog oleks võimalikult vähe mõjutatud.

---

<sup>1</sup> [sowiso.nl/en/](http://sowiso.nl/en/)

Valmis lahendusi Wordi sisu lugemiseks on palju – näiteks Component Source<sup>1</sup>, DevExpress<sup>2</sup>, Telerik<sup>3</sup>, kuid nendest ükski ei toeta matemaatiliste valemite süntaksianalüüsi.

### **2.3 Taustuuringu kokkuvõte**

Uuringu tulemusena selgus, et Word dokumendist matemaatika valemite lugemist ei toeta ükski valmis lahendus. Kuigi eksisteeris olemasolevad matemaatika ülesannete lahendamist toetavad keskkondi, siis nad ei toeta samuti sisuloomet kasutades Wordi ning ei vasta kliendi nõudele, kus soovitakse ka lisaarenduse võimekust tulevikus.

---

<sup>1</sup> componentsource.com

<sup>2</sup> devexpress.com

<sup>3</sup> telerik.com

### 3 Analüüs

Analüüsi käigus toodi välja lihtne kasutajaliidese kirjeldus, nõuded projektile ning tehtud eeltegevused kasutajaliidese arendamise jaoks.

#### 3.1 Objekti kirjeldus

Ülesannete kogumik on põhiobjekt, iga aasta luuakse uus kogumik kirjastuse poolt. Kogumiku iseloomustajad on pealkiri ja kirjeldus, mis täpsustab kogumiku sisu. Kogumike vahel peab olema lihtne navigeerida, mis tähendab, et kogumikud peavad olema jaotatud lehekülgedena ning kasutama filtreerimist. Kogumik koosneb erinevatest ülesannetest.

Ülesandega on seotud erinevad vastused, mis käituvad nagu valikvariandid, millest üks on õige. Ülesandega on veel seotud üks või mitu lahendust, mida kuvatakse, siis kui õige valikvariant valitud. Nii ülesanne, vastus kui lahendus koosneb sisust, mis võib sisaldada teksti, matemaatilisi valemeid, tabelleid ja jooniseid.

#### 3.2 Töö protsess

Koodi kirjutamiseks on kasutusel IntelliJ<sup>1</sup> IDE (ingl *integrated development environment*), mis on üks tugevamatest veebi programmeerimist toetav integreeritud arenduskeskkond [2]. Arendamisel kasutasime agiilset lähenemist. Igal nädalal oli juhendajaga tagasivaade eelmisel nädalal tehtud tööle ning arutelu tekkinud küsimuste või takistuste üle.

IT terminite tõlkimiseks eesti keelde kasutati Glosbe sõnaraamatut<sup>2</sup>, Eesti Keele Instituudi inglise-eesti sõnastikku<sup>3</sup> ning IT terministandardi sõnastikku<sup>4</sup>.

---

<sup>1</sup> [jetbrains.com/idea/](https://jetbrains.com/idea/)

<sup>2</sup> [et.glosbe.com](https://et.glosbe.com)

<sup>3</sup> [eki.ee/dict/ies/index.cgi](https://eki.ee/dict/ies/index.cgi)

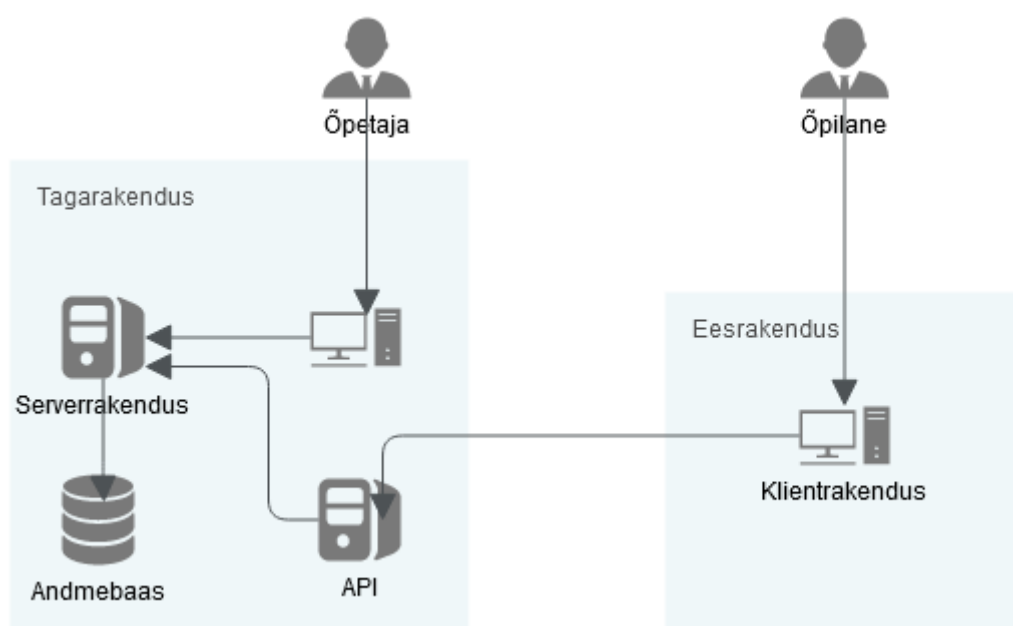
<sup>4</sup> <http://www.eki.ee/dict/its/index.cgi>



### 3.3 Üldine arhitektuur

Süsteemi rollideks on õpetaja ja õpilane. Tagarakenduseks (Joonis 1) on monoliit, kus on õpetaja jaoks algeline veebileht, kus saab ülesannete kogumikke üles laadida süsteemi. Üles laaditud sisu tarbitakse õpilaste poolt läbi eesrakenduse.

Antud lõputöös realiseeritakse eesrakendus. Eesrakendus realiseeritakse klient-serveri<sup>1</sup> arhitektuuri mudeli põhjal. Andmete kättesaamiseks suhtleb eesrakenduses asuv veebileht e klientrakendus (ingl *front-end application*) tagarakenduses asuva APIga (ingl *Application Programming Interface*).



Joonis 1. Üldine arhitektuur.

### 3.4 Funktsionaalsed nõuded

Funktsionaalsed nõuded kasutajaliidesele:

- Ülesannete kogumikke on võimalik filtreerida otsingusõnega.
- Ülesannete kogumikud on jagatud lehekülgedeks.
- Arhitektuur on mikroteenuste põhimõttel, eesrakendus suhtleb tagarakendusega kasutades API't.

<sup>1</sup> en.wikipedia.org/wiki/Client-server\_model

- Ülesandeid lahendades saab valida valikvariantide vahel, õiged ja valed vastused on eristatavad
- Peale õige vastuse valimist näidatakse kasutajale võimalikke lahenduskäike.

### 3.5 Mittefunktsionaalsed nõuded

Mittefunktsionaalsed nõuded projektile:

- Töötab laialt levinud brauserites.
- On kasutatav lauaarvutitest kuni mobiiltelefonideni.
- Veebileht ja tema suhtlus API'ga on turvaline.
- On kasutajasõbralik. Näeb välja konkreetne ning modernne.

### 3.6 Näidismudel

Kasutajaliidese nägemuse ühildamiseks kliendi ja arendaja vahel sai tehtud näidismudel (ingl *mock-up*). Näidismudeli tegemiseks kasutati kasutajaliidese disaini tööriista Figma<sup>1</sup>. Näidismudelis on välja toodud kasutajaliidese avavaade (vt Lisa 2), ülesande lahendamine valikvastustega (vt Lisa 3), lahenduskäigu kuvamine (vt Lisa 4) ning mobiilivaates avavaade (vt Lisa 5).

---

<sup>1</sup> figma.com

## 4 Klientrakenduse arendamine

Peatükis kirjeldatakse loodud klientrakenduse arenduskäiku ning tehtud otsuseid.

### 4.1 Koodi stiil

Koodi stiili reeglite kirjeldamiseks on kasutatud TSLinti<sup>1</sup>. Tagamaks koodi jätkusuutlikust ning selget kirjutamist on võetud kasutusele Robert C. Martini raamat „*Clean Code*“. Kuna kasutusel on Angular, siis on võetud arvesse ka Angulari stiili juhendit<sup>2</sup>, kus on kirjas Angulari arendajate koodi stiili nõuanded ja soovitused.

#### 4.1.1 LIFT

LIFT (ingl *Locate, Identity, Flat, T-DRY*) on neli põhi stiili soovitus Angularis, mis aitavad luua koodi, mis skaleerib hästi, on modulaarne ja suurendab arendaja efektiivsust läbi koodi kiire üles leidmise. Järgnevas tabelis (Tabel 7) on stiili juhendist välja toodud põhjendused [3].

Soovitus	Tähendus ja põhjendus
Leia (ingl <i>Locate</i> )	Tee koodi leidmine võimalikult intuitiivseks, lihtsaks ja kiireks. Seotud faile peaks hoidma koos ning loogilises asukohas.
Tuvasta (ingl <i>Identify</i> )	Vähendamaks aega koodi läbiotsimiseks, tuleks kasutada faili nimesid, mis näitavad koheselt, mis seal failis on ning mida ta teeb. Pikemad nimed on paremad kui lühikesed aga ebaselged. Välti faile, kus on mitu komponenti või segu erinevast ärioloogikast.
Lame (ingl <i>Flat</i> )	Hoia projekti struktuur võimalikult lame. Alamkaustasid luua siis, kui kaustas on seitse või rohkem faili.
Proovi ennast mitte korrata (ingl <i>Try not to repeat yourself</i> )	Ära korda ennast, kuid ära ohverda selle saavutamiseks loetavust.

Tabel 7. LIFT põhimõtted

<sup>1</sup> [palantir.github.io/tslint/](https://palantir.github.io/tslint/)

<sup>2</sup> [angular.io/guide/styleguide](https://angular.io/guide/styleguide)

## 4.2 Kasutatud tehnoloogiad

Antud alapeatükis kirjeldatakse tööriistaid ja raamistikke, mida kasutatakse klientrakenduse loomiseks.

### 4.2.1 Angular

Javascripti<sup>1</sup> raamistik on üks parimatest viisidest, kuidas parendada veebilehte arendades produktiivsust, koodi loetavust ning lihtsustada koodi ülal pidamist. Hetkel populaarsemad Javascripti raamistikud on ReactJs<sup>2</sup>, Angular<sup>3</sup> ja Vue.js<sup>4</sup> [4]. Kõik kolm raamistikku on tugevad ning oma halbade ja heade külgedega, kuid kuna töö autor on nendest kõige tuttavam Angulariga, siis sai lõplikuks valikuks just see võetud.

Angular on vabavara projekt, mida arendab ja hooldab Google<sup>5</sup> ja vabatahtlike arendajate kogukond. Angular on platvorm, mille sihtgrupp on suur hulk erinevaid brausereid ja nende versioone, hübriid mobiili raamistikud ja lauaarvutite rakendused [5]. Angulari rakendused on ehitatud puhta disaini mustrite ümber, mis rõhutab rakenduste loomist, mis on lihtsasti skaleeritavad, jätkusuutlikud, testitavad ning standardiseeritud [6].

Koostöös Microsoftiga tegi Google Typescriptist<sup>6</sup> vaikumisi keeleks Angularis. Typescript on Javascripti ülemhulk, mis tähendab, et kõik Javascripti kood on ka Typescripti kood. Samuti võimaldab Typescript arendajatelt toetada pärand brausereid nagu Internet Explorer<sup>7</sup>, samas lubades neil kirjutada modernset koodi, mis töötab brauserites nagu Chrome<sup>8</sup>, FireFox<sup>9</sup> ja Edge<sup>10</sup> [5]. Typescript toetab Javascripti lisades juurde näiteks staatilised tüübid, klassid ja liidesed. Typescript kompileeritakse Javascriptiks, kompileerimise ajal on võimalik püüda vigasid, mis võiksid jääda märkamata, kui kasutada puhast Javascripti.

---

<sup>1</sup> javascript.com

<sup>2</sup> reactjs.org

<sup>3</sup> angular.io

<sup>4</sup> vuejs.org

<sup>5</sup> google.com

<sup>6</sup> typescriptlang.org

<sup>7</sup> microsoft.com/en-us/download/internet-explorer.aspx

<sup>8</sup> google.com/chrome/

<sup>9</sup> mozilla.org/en-US/firefox/

<sup>10</sup> microsoft.com/en-us/edge

Angulariga koos tuleb kasutusele võtta tööriistad nagu Node.js<sup>1</sup> ja npm<sup>2</sup> (ingl node package manager). Veebilehel sisu näitamiseks on vaja mehhanismi, mis laeks HTML (ingl HyperText Markup Language) faili koos seonduvate CSS (ingl Cascading Style Sheets) ja Javascripti failidega. Selle jaoks kasutame Node't, kui veebiserverina, mis laeb kõik vajaliku [7]. Angular ja tema paketid sõltuvad npm pakettidest paljude tunnuste ning funktsioonide jaoks. Samuti võimaldab npm oma laia pakettide kogumikuga võtta kasutusele valmis lahendusi erinevate arendusprobleemide lahendamiseks.

#### 4.2.2 Tslint

TSLint (ingl Typescript Lint) on staatiline analüüsi tööriist, mis kontrollib Typescripti koodi loetavust, jätkusuutlikust ning funktsionaalseid vigu. TSLint toob endaga kaasa põhireegleid ning aitab automaatselt parandada vormindamise ja stiili vigu [8]. TSLint'ga saab ühtlast koodi ja struktuuri stiili tagada üle tiimi või peale arendajate vahetumist.

#### 4.2.3 SCSS ja Bootstrap

Mida suuremaks läheb projekt, seda rohkem tekib CSS ridu. Sellega kaasnevad probleemid on näiteks - tuleb näha palju vaeva, et teha väikseid muudatusi, tekib raskusi koodi restruktureerimisel, tekib üleliigset või korduvat koodi. SCSS<sup>3</sup> (ingl *Sassy CSS*) on eeltöötaja, mis aitab lahendada neid probleeme. SCSS lisab CSS'st puuduvad võimekused nagu funktsioonid, muutujad ning pesastamine (ingl *nesting*).

Teiseks abivahendiks on CSS raamistikute kasutamine. CSS raamistikud toovad endaga kaasa juba valmis klassid, mis teevad veebilehe disainimise ühtlasemaks ning väldib CSS lisaridade tekkimist. Populaarsemad raamistikud on Bootstrap<sup>4</sup>, Foundation<sup>5</sup> ja Bulma<sup>6</sup> [9]. Neil kõigil on olemas vajalik võimekus antud projekti jaoks, valiku teen Bootstrapi kasuks, sest ta on kõige laialt levinum ning teeb koodi pikaajalise haldamise lihtsamaks.

Bootstrap on laialt levinud raamistik veebilehtede disainimiseks. Olles paindlik ja väikese õpikõveraga, aitab ta veebilehti mugavamalt ning kiiremini üles seada. Bootstrap pakub

---

<sup>1</sup> [nodejs.org/en/](https://nodejs.org/en/)

<sup>2</sup> [npmjs.com](https://npmjs.com)

<sup>3</sup> [sass-lang.com](https://sass-lang.com)

<sup>4</sup> [getbootstrap.com](https://getbootstrap.com)

<sup>5</sup> [get.foundation](https://get.foundation)

<sup>6</sup> [bulma.io](https://bulma.io)

eeliseid nagu reageerivat, mobiil esimesena (ingl *mobile-first*) disaini ja laia brauserite tuge [10].

#### 4.2.4 PrimeNG

Vältimaks ratta uuesti leiutamist, saab kasutada Angulari komponentide teeki, kus juba valmis disainitud lahendused nagu nupud, sisendid ja palju muud. Populaarsemad teegid on Angular Material<sup>1</sup>, NG Bootstrap<sup>2</sup> ning PrimeNG<sup>3</sup> [11]. Nendest valisin PrimeNG, kuna ta on suurima komponentide valikuga nendest ning hästi dokumenteeritud.

#### 4.2.5 Browserlist

Browserlist<sup>4</sup> on konfiguratsioon, mis jagab toetatavaid brauserite versioone erinevate klientrakenduse tööriistade vahel, nagu näiteks Babel<sup>5</sup> või Autoprefixer<sup>6</sup>, aidates kaasa võimalikult paljude erinevate brauserite toetamisele [12]. Antud kolm tööriista tulevad vaikimisi Angulari projektiga kaasa ning aitavad vaevavabalt kaasa vajalike brauserite toe seadistamisel.

#### 4.2.6 Jasmine ja Karma

Testimisraamistikest on kasutusel Jasmine<sup>7</sup> ja Karma<sup>8</sup>. Antud raamistikud on Angulariga vaikimisi kaasas ning katavad ühiktestimise vajadused projektis.

Jasmine on üks populaarsematest Javascripti raamistikest, mida kasutatakse testide loomiseks. Raamistik võimaldab näiteks luua imiteerivaid (ingl *Mock*) objekte või spioone (ingl *Spy*). Karma kasutatakse testide jooksutamiseks. Karma võimaldab mugavat testide haldamist ning jooksutamist [13].

---

<sup>1</sup> [material.angular.io](http://material.angular.io)

<sup>2</sup> [ng-bootstrap.github.io](http://ng-bootstrap.github.io)

<sup>3</sup> [primefaces.org/primeng/](http://primefaces.org/primeng/)

<sup>4</sup> [github.com/browserslist/browserslist](http://github.com/browserslist/browserslist)

<sup>5</sup> [babeljs.io](http://babeljs.io)

<sup>6</sup> [autoprefixer.github.io](http://autoprefixer.github.io)

<sup>7</sup> [jasmine.github.io](http://jasmine.github.io)

<sup>8</sup> [karma-runner.github.io](http://karma-runner.github.io)

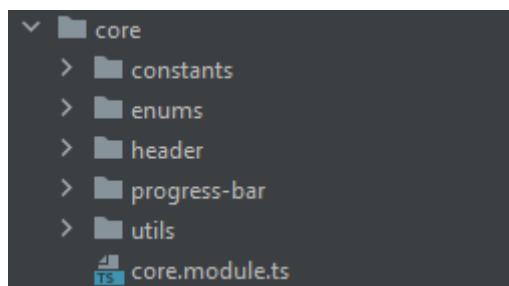
## 4.3 Projekti struktuur

Projekti struktuuri tehes lähtuti Angulari stiilijuhendist [3] ning Doguhan Uluca raamatust „*Angular for Enterprise-Ready Web Applications*“ [5].

Angulari algne genereeritud struktuur sobib hästi väikeste projektide jaoks, mida on vaja kiiresti valmis saada. Projekti kasvades või arendajate vahetustega võib aga algsest struktuurist saada arendamist aeglustav ning tehnilist võlga suurendav takistus. Selle tõttu seati antud projektis üles struktuur, mis peab silmas ka projekti hiljemt laiendamist.

Rakendus koosneb 3 põhiosast, kaustad nimega – tuum (ingl *core*), põhilood (ingl *features*) ja jagatud (ingl *shared*).

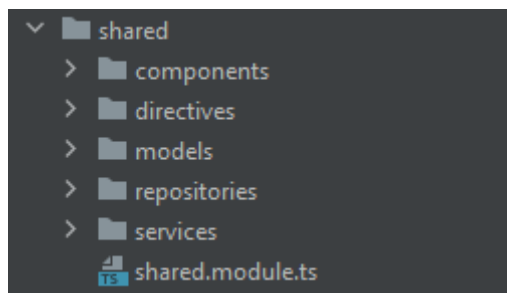
Tuum (Joonis 2) võtab endale juurmooduli rolli. Siin esinevate failide põhisarnasus on, et neid laetakse ühe korra käitusajal (ingl *runtime*) [14]. Siin asuvad staatilised komponendid nagu päis (ingl *header*), laadimisriba (ingl *progress-bar*), konstandid, loetelud (ingl *enumerations*) ja utiliidid.



Joonis 2. Tuuma kausta struktuur.

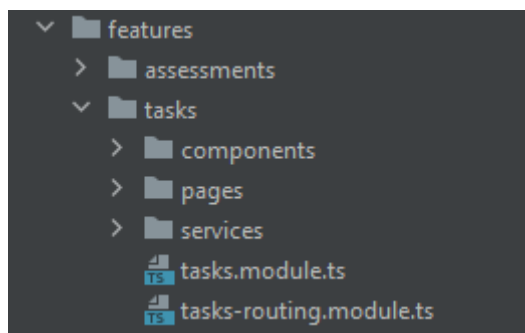
Jagatud kaust (Joonis 3) aitab asünkroonselt laadida (ingl *lazy loading*) Angulari mooduleid. See aitab vähendada versioonide (ingl *build*) suurust. Tänu sellele väheneb lisaks paigaldamise ning rakenduse esmase laadimise kiirus. Jagatud kaust ei tohiks sõltuda ülejäänud projektist, ehk ei tohiks ühtegi teist moodulit importida. Seda jälgides aitab moodul koodi paremini organiseerida ning koodi kirjutamist ladusamaks teha [14]. Jagatud kaustas saab laadida kõik vajalikud välised moodulid, mida on vaja erinevates tunnuste moodulites. Siin asuvad üldkasutatavad komponendid, direktiivid, mudelid, repositooriumid ja teenused. Mudelite kaustas asuvad API poolt tulevate andmete kirjeldatud struktuurid. Repositooriumid ning teenused on Angulari vaates mõlemad süstitavad (ingl *injectable*) teenused. Repositooriumid on eraldatud, kuna seal asuvad

teenused, mis suhtlevad API pakutavate teenustega, eraldades nende loogika rakenduse sisesest loogikast.



Joonis 3. Jagatud kausta struktuur.

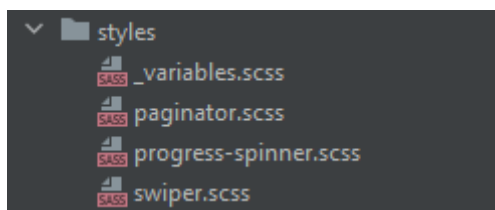
Põhiloode (ingl *features*) kaust (Joonis 4) sisaldab rakenduse põhiobjekte, nagu ülesannete kogumikud (ingl *assessments*) ja ülesanded (ingl *tasks*). Alamkaustade sees on nende spetsiifiliste põhilugudega seotud komponendid, leheküljed (ingl *pages*) või teenused. Lehekülgede kaustas on kõik erinevad vaated. Leheküljed võivad koosneda komponentidest ning neid toetavad teenused.



Joonis 4. Põhiloode kausta struktuur.

*Styles* faili lisatakse globaalset või üldist stiili koodi ning on koht Angularis, mis võib kiiresti muutuda oma paljude ridade tõttu raskesti hallatavaks. Kuigi selle parendamiseks on kasutusel SCSS, siis sadade või tuhandete ridade vältimiseks ühes failis, tuleks jagada kood loogilisteks tükkideks. Selleks tuleb lisada eraldi stiilide kaust (Joonis 5). Stiilide kaust peab asetsema SCSS puhul väljaspool varade (ingl *assets*) kausta. Kuna lõplikult on rakendusel siiski vaja kompileerimisel ainult CSS koodi, aga kogu varade kaust antakse rakendusele ehitamisel kaasa, mis tähendaks, et võib juhtuda olukord, kus rakenduse paigaldamisel on lisaks CSS'le kaasas ka SCSS failid, mida rakendusel ei ole enam vaja.





Joonis 5. Stiili kaust.

## 4.4 Klient-server kommunikatsioon

Klientrakenduse ja API suhtlus põhineb REST (ingl *Representational state transfer*) arhitektuuril, kasutab HTTP (ingl *Hypertext Transfer Protocol*) protokollil ning andmete vahetamiseks on kasutusel JSON (ingl *JavaScript Object Notation*) formaat.

### 4.4.1 RESTful API

Restful on API, mis jälgib viite põhimõtet [15]:

1. Klient-server arhitektuur - aitab vältida nende omavahelist sõltuvust. Näiteks nad ei ole omavahel tehnoloogiliselt sõltuvuses, mis võimaldab ühe serveriga ühendada mitut klienti, mis kõik on omavahel sõltumatud.
2. Olekuvaba - aitab vajadusel eskaleerida serveri infrastruktuuri, kuna iga päringuga tuleb kaasa kõik vajalik, et päringut läbi viia.
3. On võimeline kasutama päringute vahemälu hoidmist, et vähendada koormust serverile.
4. Ühtlane liides - rakenduste vaheline suhtlus tuleb lihtsa ning hallatava arhitektuuriga, kuna kõik lõpp-punktid kasutavad samu reegleid andmete vahetamisel.
5. Kihistatud – süsteem koosneb hierarhilistest kihtidest, millest komponent näeb ainult lähimat kihti, millega ta suhtleb.

Rakenduses kasutatavate lõpp-punktide kirjeldamisel kasutati RESTful disaini põhimõtteid [16] ning järgnevalt on toodud lõpp-punktid ülesannete kogumike vaatest:

1. `api/assessments`
2. `api/assessments/{:assessmentId}`
3. `api/assessments/{:assessmentId}/tasks`
4. `api/assessments?page=1&search=otsingusõne`

Kõikidel päringutel on kasutuses GET meetod, loogeliste sulge vahel on kirjeldatud muutujad. Esimene lõpp-punkt tagastab kasutajale kõik talle kättesaadavad ülesannete kogumikud. Järgmises lõpp-punktis saab pärida unikaalse identifikaatoriga ühe spetsiifilise kogumiku kirje. Kolmandas lõpp-punktis saab päringu vastuseks vastavalt identifikaatorile ülesannete kogumikuga seotud ülesanded.

Esimesel lõpp-punktis on lisaks kasutatud lehekülgedeks jagamist (ingl *paging*) ning otsingu võimalust, mille näide on demonstreeritud neljandas punktis. Otsing kasutajaliideses on olemuselt lihtne, on üks otsinguväli, mida täites otsitakse ülesannete kogumiku mudeliga seotud andmestikust. Lehekülgedeks jagamine toimub tagarakenduse pool, et mitte vahetada üleliigseid andmeid rakenduste vahel, vähendades koormust serveritele ning läbi selle kiirendades veebilehe laadimiskiirust. Klientrakenduses lehekülgedeks jagamise kasutamiseks pidi kirjeldama mudeli (Joonis 6), kus esemed (ingl *items*) on tüüpi parameetrid (ingl *generic*), ehk sinna saab asendada vajadusel ükskõik millise klassi. Lisaks on tähtsad lehekülje suurus (ingl *page size*), mis näitab esemete kogust praegusel leheküljel ning kogu lehtede arv (ingl *total pages*), mis aitab kasutajale näidata tema võimalikke navigeerimisvõimalusi.

```
export interface PageableResponse<T> {  
  
    items: T[];  
  
    pageSize: number;  
  
    totalPages: number;  
  
    hasNextPage: boolean;  
  
}
```

Joonis 6. Lehekülgedeks jagamise mudel.

## 4.5 Konfiguratsioon

Konfiguratsioon rakenduses on jaotatud kaheks. Vastavalt keskkonnale - tootmine või lokaalne keskkond. Konfiguratsioonis on määratud rakenduse logimistase, tootmises on tase miinimum, mis vajalik. Samuti on määratud API aadress vastavalt keskkonnale ning tootmise lipp.

Kui produktsiooni lipu väärtus on tõene, siis optimeeritakse rakendust rohkem kui tavaliselt. Enam ei jäeta faile loetavale kujule, vaid minimiseeritakse ning kasutakse lühemaid funktsiooni ja muutujate nimesid, mis muudab koodi loetamatumaks, aga vähendab ehituspaki (ingl *build*) suurust. Lisaks eelkompileeritakse kood, mille käigus eemaldatakse kõik kasutusel mitte olev kood [17].

## 4.6 Ülesannete kuvamine

Ülesannete kuvamisel sai kõigepealt katsetatud Wordi valemite jagamist piltideks. Et valemid jääksid ühtse osana tekstist või ülesandest, mitte eraldi välja tooduna, oli vaja teha valemite, jooniste, tabelite või teksti koos kuvamiseks igast ülesandest või lahendusest eraldi PNG (ingl *Portable Network Graphics*). Selline lahendusviis osutus ebamugavaks ja ebaefektiivseks lahenduseks. Pilte on raskem teha loetavaks kõikidel ekraanidel, kuna teksti suurust eraldi muuta ei saa, vaid ainult pilti sisse suumida. Pildi kvaliteet jäi samuti soovida, kvaliteeti suurendades aga suureneb ka pildi salvestamiseks kuluv mälu ning edastamisel vajalik ribalaius. Valemite piltideks tegemise asendamiseks sai võetud kasutusele MathML (ingl *Mathematical Markup Language*).

MathML on mõeldud matemaatilise ja teadusliku sisu esitamiseks veebis. MathML saab kasutada visuaalselt kõrge kvaliteediga matemaatiliste valemite kuvamiseks. MathML on madala taseme keel ning ei ole mõeldud käsitsi kirjutamiseks või muutmiseks. Olles XML'i (ingl *Extensible Markup Language*) dialekt, saavad brauserid põliselt toetada MathML'i, kuigi paljud populaarsemad brauserid seda võimalust ei ole kasutanud [18]. Kuna eesmärk oleks ikkagi toetada võimalikult palju brausereid, siis peab kasutusele võtma teegi, mis võtaks sisendina MathMLi ja samas toetaks erinevaid brausereid.

Eesmärgi saavutamiseks on olemas kaks sobivat teeki – MathJax<sup>1</sup> ja KaTeX<sup>2</sup>. Mõlematel on olemas sobiv funktsionaalsus, et kuvada vajavat sisu, MathJax 3 on peaaegu sama kiire viimistlemises (ingl *rendering*) kui KaTeX, aga MathJax toetab laiemat valikut võrrandeid, mida ta suudab kuvada [19]. MathJax sai valitud viimase mainitud eelise tõttu.

---

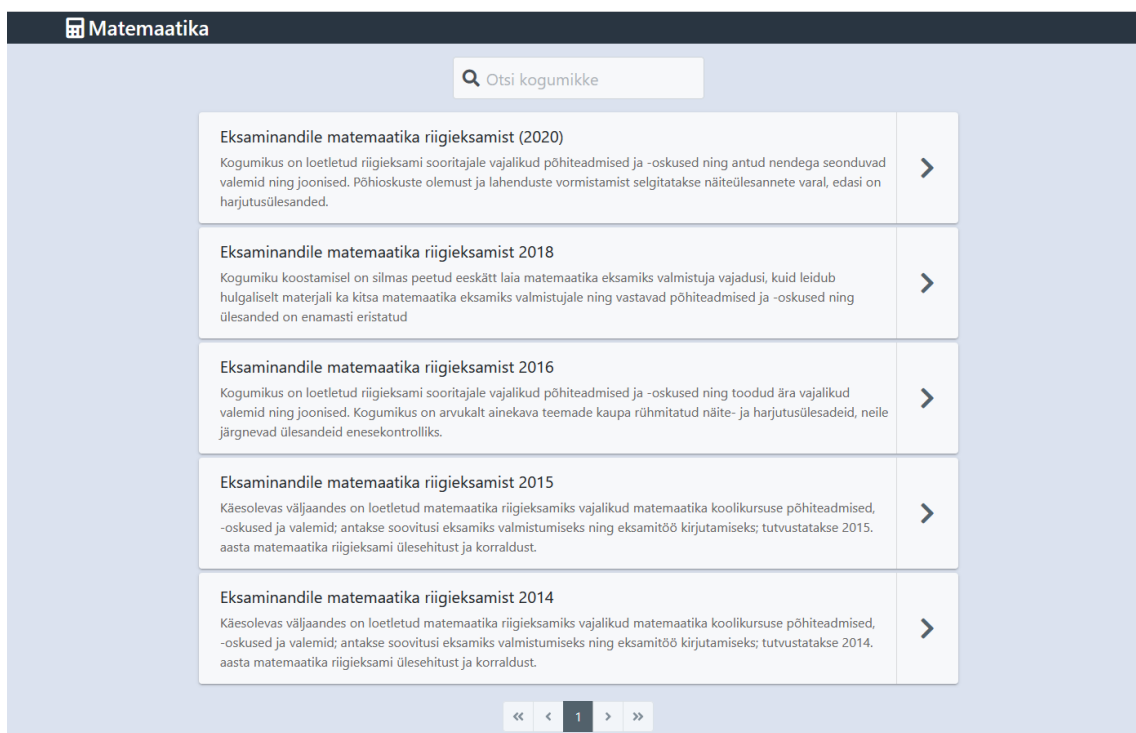
<sup>1</sup> mathjax.org

<sup>2</sup> katex.org

Lõplikus lahendis tagarakendus võtab sisendina Wordi dokumendi, sealt loetakse sisse ülesanded, lahendid ja lahendusvariandid. Kuna Word ise koosneb XML formaadist, siis seda on lihtne MathML'ks ümber teisendada. Näiteks ülesanne salvestatakse HTML formaadis, millele on vajadusel lisatud pilt või tabel, ning valemid on mähitud MathML formaati. Andes edasi ülesande sellisel kujul MathJaxile, renderdab ta MathML formaadist kuvamiseks valmis valemid.

## 4.7 Vaated

Rakendus on kättesaadav aadressil <https://matemaatika.taltech.ee>. Veebilehele minnes avaneb kasutajale avaleht (vt Joonis 7). Avalehel on kogumike nimekiri, mis on jagatud lehekülgedeks ning filtreerimise võimalus, kus otsitakse sisendit kogumike pealkirjast või kirjeldusest.



Joonis 7. Kasutajaliidese avavaade

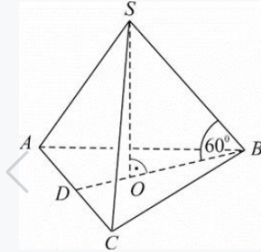
Klõpsates kogumikule avaneb sellega seotud ülesannete üldvaade (vt Joonis 8). Üldvaates kuvatakse kasutaja jaoks lihtsa ülevaatenähtena kõiki ülesandeid, millest ta saab valida endale harjutamiseks sobiva sisuga ülesande.

Matemaatika	
< Kogumikud	Eksaminandile matemaatika riigeksamist 2018
Mitu neljakohalist arvu, milles korduvaid numbreid ei esine, saab moodustada numbritest 0, 1, 2, 3?	>
Mitu erinevat kolmekohalist arvu, milles korduvaid numbreid ei esine, saab moodustada numbritest 1, 2, 3, 4, 5, ?	>
Mitu numbrite poolest üksteisest erinevat kolmekohalist arvu saab moodustada numbritest 1, 2, 3, 4, 5, kui arvus ükski number ei kordu.	>
Koosviibimise algul kõik osavõtjad käitlevad üksteist. Kokku vahetati 210 käepigistust. Mitu inimest osales sellel koosviibimisel?	>
Keegi ilmatark täheldas, et tema kodukohas on ilm 60% tõenäosusega vihmane ja 40% tõenäosusega ilus (kuiv ja selge), kui eelmisel päeval oli vihmane ilm. Seevastu, kui eelmisel päeval oli ilus ilm, siis järgmisel päeval on kõigest 20% tõenäosusega vihmane ja 80% tõenäosusega ilus ilm. Leidke tõenäosus, et ülehommene on ilus ilm, kui täna on ilus ilm.	>
Kotis on 40 ühesugust kuulikest, mis kannavad numbreid 1 - 40. Kotist võetakse juhuslikult üks kuulike. Kui suur on tõenäosus, et väljavõetud kuulikele kirjutatud arvu jagamisel kolmega tekib jääk 1?	>

Joonis 8. Kogumiku ülesannete nimekiri.

Valides kindla ülesande avaneb selle ülesande lahendamiseks ülesanne koos valikvariantidega (vt Joonis 9). Ülesannete vahel navigeerimisel on võimalik kasutada nipsamist (ingl *swipe*), või kasutada nooli. Valikvastused asuvad ülesande all, millest üks on õige. Vale variandi puhul muutub vastus punaseks, koos kaasneva tekstiga, et aidata kaasa värvipimedatel kasutajatel ülesannete lahendamist.

Korrapärase kolmnurkse püramiidi põhiserv on  $a\sqrt{3}$  ning nurk külgserva ja põhja vahel on  $60^\circ$ . Avaldage püramiidi ruumala.



Sirge ja tasandi vaheliseks nurgaks nimetatakse nurka selle sirge ja tema ristprojektsiooni vahel tasandil.

Püramiidi  $SABC$  külgserva  $SB$  ristprojektsioon põhjal on lõik  $OB$ , järelikult  $\angle SBO = 60^\circ$ .

Kolmnurk  $ABC$  on võrdkülgne kolmnurk,  $AB = BC = CA = a\sqrt{3}$ .

$\frac{9a^3}{8}$

$V_5^3 = \frac{5!}{2!} = 60$

$\frac{\sqrt{6}}{2} a^3$

Vale!

$\frac{\sqrt{3}}{4} a^3$

Joonis 9. Kogumiku ülesande vaade koos valikvastustega

Õige vastuse valimisel muutub vastus roheliseks, ülejäänud vastused peidetakse ning näidatakse ülesandega seotud võimalikke lahenduskäike (vt Joonis 10).

$S_l$  – põhjaga paralleelse lõike pindala,  
 $S_p$  – põhja pindala,  
 $h_l$  – põhjaga paralleelse lõike kaugus tipust,  
 $h$  – püramiidi kõrgus.

35 cm

Lahendus 1   Lahendus 2

Kui püramiidi põhjaga paralleelse lõike kaugus püramiidi põhjast on 14 cm, siis lõike kaugus püramiidi tipust on  $h_l = (h - 14)$ cm.

Püramiidi põhjaga paralleelse lõike omaduse põhjal saame võrde

$$\frac{54}{150} = \frac{(h-14)^2}{h^2}.$$

Võrdetegur  $k = \frac{54}{150} = 0,36$ .

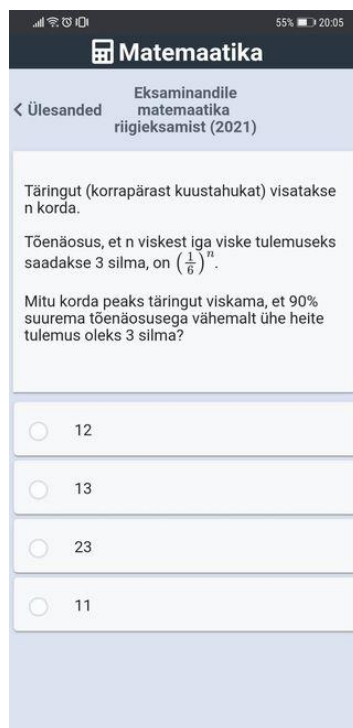
Seega  $\frac{(h-14)^2}{h^2} = 0,36 \Rightarrow \frac{h-14}{h} = \pm 0,6 \Rightarrow 1) \frac{h-14}{h} = +0,6 \Rightarrow h = 35,$

$$2) \frac{h-14}{h} = -0,6 \Rightarrow h = 8,75 < 14 \text{ (võõrlahend).}$$

Püramiidi kõrgus on 35 cm.

Joonis 10. Ülesande vaade koos lahenduskäikudega

Kõik vaated reageerivad pikslite arvu või ekraani suuruse muutustele, muutes elemendid sobivaks suuruseks. Joonis 11 demonstreerib ülesande vaadet koos valikvastustega mobiili ekraanil.



Joonis 11. Ülesande detail vaade mobiilis.

## 4.8 Turvalisus

Rakenduse arenduses on arvestatud Angulari arendajate turvasoovitustega [20]. Alapeatükis on välja toodud seadistused lisaks Angulari vaikimisi rakendatud kaitsemeetmetele.

### 4.8.1 HTTPS

HTTPS (ingl *Hypertext Transfer Protocol Secure*) on HTTP krüptitud versioon, mida kasutatakse andmete edastamiseks üle interneti. HTTPS kaitseb saadetavate andmete terviklikkust, konfidentsiaalsust ja kaitseb vahendajaründe (ingl *man-in-the-middle attack*) eest. Terviklikust kaitsedes saab vältida olukordi, kus andmed on muutunud või korrumpeerunud tahtlikult või mitte, ilma, et see oleks arusaadav [21].

Rakenduses on kasutusele võetud HTTPS protokoll kasutades Azure CDN (ingl *Content Delivery Network*) lõpp-punkti. Kõigepealt registreeriti aladomeen Taltech'is, mille CNAME (ingl *Canonical Name*) sissekanne suunab lehte CDN lõpp-punkti suunas. Antud lahendusega sai Azure poolt seadistada veebilehele TLS (ingl *Transport Layer Security*) sertifikaadi ning suunata kogu HTTP liiklus automaatselt HTTPS protokollile.



## 4.8.2 CORS

CORS (ingl *Cross-Origin Resource Sharing*) on brauseri mehhanism, mis piirab ligipääsu ressurssidele, mis asuvad väljaspool antud domeeni. Tagarakendusse päringut tehes lisatakse päritolu HTTP päringu päisesse ning serveri konfiguratsiooni põhjal otsustakse, kas antud päritolu on lubatud [22]. Eesrakenduses CORS on kasutusel ning tagarakenduse poolel on tehtud seadistused, lubamaks neil omavahel suhelda vastavalt keskkonnale.

## 4.9 Erinevad ekraanid

Erinevate ekraani suuruste toetamiseks on kasutusel Bootstrapi valmis klassid. Mobiil esimesena disainimuster võimaldab lihtsalt kogu lehekülge teha reageerivaks ekraani suurusele ning pikslite arvule, ilma, et peaks tegema uut kasutajaliidest näiteks telefonirakenduse näol. Väiksemate ekraanide puhul on tehtud kindlaks, et elemendid ei oleks liiga väikesed ning navigeerimine oleks võimalikult mugav. Rakenduses kasutatud komponendid toetavad kõiki modernseid brausereid. Ülesannete lahendamise vaates on kasutusel Swiper<sup>1</sup>, mis võimaldab ülesannete vahel navigeerida kasutades nipsamist (ingl *swipe*).

## 4.10 Rakenduse paigaldamine

Rakenduse paigaldamiseks (ingl *deploy*) on vaja kõigepealt vaja failid valmis ehitada. Produktsiooni keskkonna failide ehitamiseks on kasutusel käsklus „`ng build --prod`“. Loodud artefaktid (ingl *artifacts*) asuvad `/dist` kaustas. Kausta sisu paigaldamiseks on käsklus „`ng deploy`“. Kasutades pakki `@azure/ng-deploy`<sup>2</sup>, on paigaldamise protsess Azures tehtud automaatseks.

---

<sup>1</sup> swiperjs.com

<sup>2</sup> npmjs.com/package/@azure/ng-deploy

## 5 Kasutajaliidese valideerimine

Arendatud kasutajaliidese funktsionaalsuse valideerimiseks hinnatakse ning kontrollitakse rakendust vastu analüüsi käigus seatud nõutele.

### 5.1 Brauserid ja erinevad seadmed

Vastavalt statistikale, 2021. aasta Märtsi seisuga, on Eestis levinuimad brauserid Chrome, Safari<sup>1</sup>, Firefox, Opera<sup>2</sup>, Edge ning Samsung Internet<sup>3</sup>, millega katab >98% kasutavast brauseritest [23]. Brauseri või seadme valideerimiseks tehakse manuaalne testimine, mille käigus avatakse rakenduse veebileht, kus proovitakse otsida, navigeerida ülesannete ja kogumike vahel ning ülesandeid lahendada. Jälgitakse, et kasutajaliidese komponendid ei oleks paigast ära, tekst loetav, matemaatilised valemid kuvatud ning rakendus oleks kasutatav ilma takistusteta. Erinevate ekraani suuruste simuleerimiseks on kasutusel Firefox'i arendustööriist, kus saab määrata ekraanisuuruseid. Järgnevas tabelis on toodud välja seadmed ning nendes testitud brauserid koos versioonidega (vt Tabel 8).

Tabel 8. Testimiseks kasutatud seadmed ning brauserid.

Seade	Operatsiooni süsteem	Testitud brauserid
PC (ingl <i>Personal Computer</i> )	Windows 10	Chrome v90, Firefox v88, Edge v90, Opera v76
Huawei P20 Mate	Android v10	Chrome v90, Samsung Internet v14, HUAWEI Browser <sup>4</sup> v11
iPhone 8	iOS v14	Safari v14
Samsung Galaxy S5	Android v6	Samsung Internet v10

Kõik testitud brauserid ja seadmed läbisid manuaalse testimise edukalt. Lisaks neile kontrolliti üle siit puuduolevaid ekraani suuruseid nagu näiteks tahvelarvuti või keskmine sülearvuti. Kasutajaliides oli kõikidel ekraanidel reageeriv ning kasutajakogemuses

<sup>1</sup> [apple.com/safari/](http://apple.com/safari/)

<sup>2</sup> [opera.com](http://opera.com)

<sup>3</sup> [samsung.com/us/support/owners/app/samsung-internet](http://samsung.com/us/support/owners/app/samsung-internet)

<sup>4</sup> [consumer.huawei.com/en/mobileservices/browser](http://consumer.huawei.com/en/mobileservices/browser)

probleeme ei olnud, välja arvatud üksikud juhud, kus väiksematel ekraanidel matemaatilised valemid ei jäänud ekraani raamidesse.

## 5.2 Üksustestid

Rakenduse töökindluse suurendamiseks ning edasiarendamise hõlbustamiseks on rakenduse tähtsam äriloogika kaetud üksustestidega. Testidega on kaetud komponendid, mis töötlevad andmeid või mõjutavad, mis andmeid kuvatakse. Komponente, mis lihtsalt vaheldavad andmeid ei ole üksustestidega kaetud. Projektis käsurealt jooksutades „ng test“ jooksutatakse Karma poolt kõik projektis asuvad testid automaatselt. Tulemused kuvatakse nii käsureal kui ka brauseris, mis avatakse Karma poolt, kus kuvatakse loetavamal viisil testide jooksutamise tulemused.

## 5.3 Turvalisus

Antud peatükis on analüüsitud klientrakenduse turvalisust põhiliselt OWASP (ingl *Open Web Application Security Project*) Top 10 standardi<sup>1</sup> vaatest. Välja toodud on need punktid, mida peab arvestama klientrakenduse turvalisuse seisukohast.

### 5.3.1 XSS

XSS (ingl *Cross Site Scripting*) on rünnaku tüüp, kus pahatahtlik skript süstitakse (ingl *injected*) muidu usaldusväärsesse veebilehte. XSS rünnak juhtub kui ründaja kasutab klientrakendust, et saata pahatahtlikku koodi, tavaliselt läbi brauseri, teisele veebilehe kasutajale. XSS on üks enimlevinumaid rünnakuid veebilehel ning võib juhtuda igal pool, kus klientrakendus kasutab kasutaja sisendit väljundis, mida ta genereerib. Tagajärgedeks võivad olla näiteks kasutaja sessiooni kaaperdamine, veebilehe sisu moonutamine või kasutajate suunamine pahatahtlikele veebilehtedele [24].

Süsteematiselt XSS rünnakute blokeerimiseks, kohtleb Angular vaikimisi kõiki väärtuseid ebausaldusväärseks. Väärtuse sisestamisel kasutaja poolt, saniteeritakse (ingl *sanitize*) ja muundatakse (ingl *escape*) väärtused ohutuks. Angulari mallid (ingl *templates*) on vaikimisi usaldatud ning neid peaks kohtlema koodina, mida saab

---

<sup>1</sup> [owasp.org/www-project-top-ten](https://owasp.org/www-project-top-ten)

jooksutada [25]. Klientrakenduses on tehtud kindlaks, et kasutaja sisendiga ei ole võimalik malle (ingl *template*) manipuleerida. Samuti ei ole võimalik kasutajal luua sisu, kõik suhtlus API'ga on läbi GET päringute, mis ei lisa ega muuda andmebaasi andmeid.

### 5.3.2 SQL süstimine

Kuna GET päringud on ainuke viis API'ga suhtluses, siis päringute ründevektoreid on vähem. Põhinõrkuseks on otsing, kus saadetakse tagarakendusse sõne, mille järgi hakatakse andmebaasist vasteid otsima. Alljärgnev koodinäide demonstreerib, kuidas saada kõik vasted alati tagasi, kuna 1+1 on alati tõese väärtusega, kui tagarakendus ei saniteeri sissetulevaid väärtuseid korralikult.

```
/api/assessments?page=1&search=sõne'+OR+1=1--
```

Päringut on võimalik modifitseerida pahatahtlikumaks ning saada andmetele ligi, millele ei tohiks olla ligipääsu. Antud juhul sisendi saniteerimine jääb tagarakenduse vastutada ja manuaalse testimise käigus ei täheldatud, et sisend jääks saniteerimata.

### 5.3.3 Komponendid teadaolevate haavatavustega

Komponendid nagu teegid või raamistikud jooksevad sama õigustega nagu rakendus ise. Haavatavat komponenti ära kasutades on võimalik andmeid varastada või nõrgestada rakenduse kaitsevõimet. Vastavalt OWASP nõuetele on rakenduses jälgitud komponentide kasutamist [26]. Rakendusest on eemaldatud kasutamata komponendid. Kasutatud on aktiivselt hoolduses olevaid komponente, millele tehakse pidevalt turvauuendusi. Kõik komponendid on arenduse hetkel viimaste turvauuendustega. Npm võimaldab *package.json* failis eesliitena tildet või sisestusmärki (ingl *caret*) kasutades teha automaatselt pakkeid rakendust mitte lõhkuvaid uuendusi. Samas peab sellist lahendust kasutades arvestama, et seda ei kasutaks vähem usaldatud pakkeid. Selle tõttu on eesliiteid kasutatud otseselt Angulariga seotud pakkeid.

### 5.3.4 Ebapiisav logimine ja seire

Ebapiisav logimine ja seire koos puuduva intsidenti lahendamise võimekusega võimaldab ründajal järjepidevalt otsida rakenduses nõrkuseid või kasutada leitud haavatavust ründeplatvormina teise süsteemi. Uuringud on näidanud, et rakenduse kaitsest läbi murdmine (ingl *breach*) avastatakse keskmiselt 191 päeva pärast juhtumit ning

avastatakse läbi kolmandate osapoolte, mitte läbi logimise või seire [27]. Antud rakenduses on logimine minimaalne ning seire olematu, edasiarendustes peaks kindlasti antud punktile rohkem tähelepanu pöörama.

## 6 Kokkuvõte

Töö eesmärgiks oli luua matemaatiliste ülesannete lahendamise keskkond, mis kuvab läbi Wordi sisestatud ülesandeid ja matemaatilisi valemeid. Kasutajaliides pidi toetama erinevaid brausereid ning erinevaid ekraani suuruseid.

Alustati olemasolevate lahenduste uurimisega, et võimalusel võtta kasutusele eesmärgi täitmiseks valmis lahendus või integreerida mõni osaliselt sobiv lahendus. Uuringu tulemusena selgus, et olemasolevad lahendused ei sobi kokku piisaval määral seatud kriteeriumitega. Eesmärgi täitmiseks analüüsiti kliendi poolt nõudeid, millele loodav rakendus peab vastama. Lisaks kirjeldati üldist arhitektuuri ning tehti näidismudel.

Töö tulemusena valmis kasutajaliides, kus kasutaja saab valida erinevate ülesannete kogumike vahel ning sobiva ülesande leidmisel saab valikvastuste abil lahendada ülesandeid. Ülesannetes on kuvatud kõik valemid, tabelid ning pildid. Rakendus on kasutatav nõutavates seadmetes ja erinevate suurustega ekraanidel ning tähtsamates kohtades üksustestidega kaetud. Analüüsi käigus püstitatud nõuded on valideeritud. Valideerimise käigus tuli välja koht, kus saaks turvalisust parendada ning väikestel ekraanidel harva esinev viga.

## Kasutatud kirjandus

- [1] „Eksaminandile matemaatika riigieksamist (2020),“ Argo Kirjastus, [Võrgumaterjal]. Available: <https://argokirjastus.ee/toode/eksaminandile-matemaatika-riigieksamist-2020/>. [Kasutatud 05 04 2021].
- [2] J. Mckinnon, „The 10 Best IDE for Web Development,“ WebsiteSetup, 16 12 2020. [Võrgumaterjal]. Available: <https://websitesetup.org/best-ide-software/>. [Kasutatud 11 05 2021].
- [3] „Angular coding style guide,“ Angular, [Võrgumaterjal]. Available: <https://angular.io/guide/styleguide>. [Kasutatud 13 04 2021].
- [4] N. Mali, „Top 10 Most Popular JavaScript Frameworks for Web Development,“ Geeks for Geeks, [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/top-10-most-popular-javascript-frameworks-for-web-development/>. [Kasutatud 07 04 2021].
- [5] D. Uluca, Angular for Enterprise-Ready Web Applications - Second Edition, Packt Publishing, 2020.
- [6] A. Freeman, Pro Angular 9: Build Powerful and Dynamic Web Apps, Apress, 2020.
- [7] B. Bachina, „How To Develop and Build Angular App With NodeJS,“ Medium, 11 03 2020. [Võrgumaterjal]. Available: <https://medium.com/bb-tutorials-and-thoughts/how-to-develop-and-build-angular-app-with-nodejs-e24c40444421>. [Kasutatud 07 04 2021].
- [8] „TSLint,“ Palantir, [Võrgumaterjal]. Available: <https://palantir.github.io/tslint/>. [Kasutatud 07 04 2021].
- [9] A. Pattakos, „9 Best CSS Frameworks in 2021,“ aThemes, 03 01 2021. [Võrgumaterjal]. Available: <https://athemes.com/collections/best-css-frameworks/>. [Kasutatud 27 04 2021].
- [10] T. Bacinger, „A simple Bootstrap tutorial,“ Toptal, [Võrgumaterjal]. Available: <https://www.toptal.com/front-end/what-is-bootstrap-a-short-tutorial-on-the-what-why-and-how>. [Kasutatud 08 04 2021].
- [11] R. Pathan, „7 Best Angular Component Libraries to Use in 2020,“ DZone, 17 11 2020. [Võrgumaterjal]. Available: <https://dzone.com/articles/7-best-angular-component-libraries-to-use-in-2020>. [Kasutatud 08 04 2021].
- [12] „What tools and how uses Browserlist,“ nicedoc, [Võrgumaterjal]. Available: <https://nicedoc.io/browserslist/browserslist-example>. [Kasutatud 08 04 2021].
- [13] S. G. d. Rosa, „Angular: Unit Testing Jasmine, Karma,“ Medium, 30 11 2017. [Võrgumaterjal]. Available: <https://medium.com/swlh/angular-unit-testing-jasmine-karma-step-by-step-e3376d110ab4>. [Kasutatud 05 05 2021].
- [14] S. Nath, „Angular File Structure and Best Practices (that help to scale),“ Medium, 25 04 2020. [Võrgumaterjal]. Available: [https://medium.com/@shijin\\_nath/angular-right-file-](https://medium.com/@shijin_nath/angular-right-file-)

- structure-and-best-practices-that-help-to-scale-2020-52ce8d967df5. [Kasutatud 13 04 2021].
- [15] „What is REST,“ RESTful API, [Võrgumaterjal]. Available: <https://restfulapi.net/>. [Kasutatud 18 04 2021].
- [16] „How to design a REST API,“ RESTful API, [Võrgumaterjal]. Available: <https://restfulapi.net/rest-api-design-tutorial-with-example/>. [Kasutatud 18 04 2021].
- [17] K. Talsania, „Angular Dev vs Prod Build,“ Medium, 20 07 2018. [Võrgumaterjal]. Available: <https://medium.com/@kavisha.talsania/angular-development-vs-production-build-671b7fd5dbf4>. [Kasutatud 18 04 2021].
- [18] „What is MathML?,“ W3C, [Võrgumaterjal]. Available: <https://www.w3.org/Math/whatIsMathML.html>. [Kasutatud 19 04 2021].
- [19] M. Bourne, „KaTeX and MathJax Comparison Demo,“ IntMath, 29 04 2020. [Võrgumaterjal]. Available: <https://www.intmath.com/cg5/katex-mathjax-comparison.php>. [Kasutatud 19 04 2021].
- [20] „Security,“ Angular, [Võrgumaterjal]. Available: <https://angular.io/guide/security>. [Kasutatud 08 05 2021].
- [21] „How To Secure your site with HTTPS,“ Google Search Central, [Võrgumaterjal]. Available: <https://developers.google.com/search/docs/advanced/security/https>. [Kasutatud 28 04 2021].
- [22] „Cross-Origin Resource Sharing (CORS),“ MDN Web Docs, 16 03 2021. [Võrgumaterjal]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>. [Kasutatud 28 04 2021].
- [23] „Browser market share Estonia,“ statcounter, [Võrgumaterjal]. Available: <https://gs.statcounter.com/browser-market-share/all/estonia>. [Kasutatud 21 04 2021].
- [24] KirstenS, „Cross Site Scripting (XSS) Software Attack,“ OWASP, [Võrgumaterjal]. Available: <https://owasp.org/www-community/attacks/xss/>. [Kasutatud 22 04 2021].
- [25] „Security,“ Angular, [Võrgumaterjal]. Available: [angular.io/guide/security](https://angular.io/guide/security). [Kasutatud 22 04 2021].
- [26] „Using components with known vulnerabilities,“ OWASP, [Võrgumaterjal]. Available: [https://owasp.org/www-project-top-ten/2017/A9\\_2017-Using\\_Components\\_with\\_Known\\_Vulnerabilities](https://owasp.org/www-project-top-ten/2017/A9_2017-Using_Components_with_Known_Vulnerabilities). [Kasutatud 23 04 2021].
- [27] „Insufficient logging and monitoring,“ OWASP, [Võrgumaterjal]. Available: [https://owasp.org/www-project-top-ten/2017/A10\\_2017-Insufficient\\_Logging%2526Monitoring](https://owasp.org/www-project-top-ten/2017/A10_2017-Insufficient_Logging%2526Monitoring). [Kasutatud 23 04 2021].



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

Mina, Kristjan Junus

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Gümnasistide matemaatika ülesannete iseseisva lahendamise keskkonna kasutajaliidese arendamine", mille juhendaja on Gunnar Piho
  - 1.1 reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2 üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

17.05.2021

---

<sup>1</sup> Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise 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 – Näidismudeli avavaade

Matemaatika

Otsi kogumikke

**Kogumik 1**  
Mitu neljakohalist arvu, milles korduvaid numbreid ei esine, saab moodustada numbritest 0, 1, 2, 3?

**Kogumik 2**  
Keegi ilmatark täheldas, et tema kodukohas on ilm 60% tõenäosusega vihmane ja 40% tõenäosusega ilus (kuiv ja selge), kui eelmisel päeval oli vihmane ilm.

**Kogumik 3**  
Koošiibimise algul kõik osavõtjad kätlevad üksteist. Kokku vahetati 210 käepigistust. Mitu inimest osales sellel koošiibimisel?

← 1 2 3 4 5 →

## Lisa 3 – Näidismudeli ülesande lahendamise vaade valikvastustega

Matemaatika

3/5

Kogumik 3

Keegi ilmatark täheldas, et tema kodukohas on ilm 60% tõenäosusega vihmane ja 40% tõenäosusega ilus (kuiv ja selge), kui eelmisel päeval oli vihmane ilm.

Seevastu, kui eelmisel päeval oli ilus ilm, siis järgmisel päeval on kõigest 20% tõenäosusega vihmane ja 80% tõenäosusega ilus ilm.

Leidke tõenäosus, et ülehomme on ilus ilm, kui täna on ilus ilm.

28%

88%

66%

35%

Tagasi Edasi

## Lisa 4 – Näidismudeli lahenduskäigu vaade

### Lahenduskäik

Ülesandes antakse ilmale kaks hinnangut: ilus (kuiv ja selge) ning vihmane. Kirjeldame kõiki ülesandes vaadeldavaid võimalusi puu abil. Fikseerime puu tipu „täna on ilus ilm“, joonestame puu harud, mis kujutavad võimalikku ilma homme ja ülehomme, märgime nende juurde vastavad tõenäosused (kümnnendmurdudena), millega on oodata vastaval harul nimetatud sündmuse toimumist. Liikudes piki iga meid huvitava puu haru puu tipust kuni selle lõpuni, leiame tõenäosuste korrutamise lause abil iga meid huvitava üksiksündmuse tõenäosuse. Märgime, antud ülesandes vaadeldavad üksiksündmused on üksteist välistavad. Lõpuks, sündmus „ülehomme on ilus ilm“ võib toimuda kas nii, et „täna ilus ilm, homme ilus ilm, ülehomme ilus ilm“ või nii „täna ilus ilm, homme vihmane ilm, ülehomme ilus ilm“. Järelikult, sündmuse „ülehomme on ilus ilm“ tõenäosuse leidmiseks tuleb kasutada tõenäosuste liitmise lauset.

Tagasi

Edasi

## Lisa 5 – Näidismudeli mobiilivaates avavaade

### Matemaatika

---

#### Kogumik 1

---

Mitu neljakohalist arvu, milles korduvaid numbreid ei esine, saab moodustada numbritest 0, 1, 2, 3?

#### Kogumik 2

---

Keegi ilmatark täheldas, et tema kodukohas on ilm 60% tõenäosusega vihmane ja 40% tõenäosusega ilus (kuiv ja selge), kui eelmisel päeval oli vihmane ilm.

#### Kogumik 3

---

Koosviibimise algul kõik osavõtjad kätlevad üksteist. Kokku vahetati 210 käepigistust. Mitu inimest osales sellel koosviibimisel?

← 1 2 3 4 5 →