

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Kaur Taal**

**Veebirakenduse loomine**  
**kursuse „Programmeerimine“**  
**programmeerimisülesannete haldamiseks**  
**Bakalaureusetöö (9 EAP)**

Juhendajad: Tauno Palts ja Reimo Palm

Tartu 2023

## **Veebirakenduse loomine kursuse „Programmeerimine“ programmeerimisülesannete haldamiseks**

**Lühikokkuvõte:** Selle bakalaureusetöö tulemusena valmis Tartu Ülikooli kursusele „Programmeerimine“ (LTAT.03.001) uus veebirakendus. Veebirakenduse abil on kursuse õppejõududel võimalik süsteemi lisada kursusega seotud dokumente. Neid dokumente on võimalik muuta, kustutada ja alla laadida. Enne rakenduse arendamist analüüsiti nii veebirakenduse olemust kui ka tehnoloogiaid, mille abil rakendust arendama hakati. Lisaks tehnoloogiate analüüsile anti ülevaade valmiva süsteemi nõuetest. Pärast arenduskäiku kirjeldati valminud osade arhitektuuri ja nende ülesandeid süsteemis. Viimase sammuna koostati kasutajate jaoks tagasisideküsitlus, mille abil hinnati süsteemi vastavust püstitatud nõuetele. Tagasisideküsitluse tulemusi kasutati osaliselt ka edasiste arendusplaanide paika panemiseks.

**Võtmesõnad:** Veebirakendus, dokumendid, FURPS.

**CERCS:** P175 Informaatika, süsteemiteooria.

## **Development of a web application for the course „Computer Programming“ for the management of programming tasks materials**

**Abstract:** As a result of this bachelor's thesis, a new web application was completed for the University of Tartu course „Computer Programming“ (LTAT.03.001). With the help of the web application, lecturers of the course will be able to add documents related to the course to the system. These documents can be edited, deleted and downloaded. Before the development of the application, both the nature of the web application and the technologies by which the application was developed were analysed. In addition to the analysis of technologies, an overview of the requirements of the completed system was provided. After the development process, the architecture of the completed parts and their functions in the system were described. As a final step, a feedback survey was conducted for users to assess the system's compliance with the established requirements. The results of the feedback survey were also partially used to establish further development plans.

**Keywords:** Web application, documents, FURPS.

**CERCS:** P175 Informatics, system theory.

## Sisukord

Sissejuhatus .....	4
1. Veebirakendus .....	5
1.1 Versioonihaldussüsteem .....	5
1.2 Populaarsed andmebaasid .....	5
1.3 Raamistikud .....	6
1.3.1 Angular .....	6
1.3.2 Spring Boot .....	7
2. Rakendus .....	9
2.1 Nõuded .....	9
2.2 FURPS tehnika .....	9
2.2.1 Funktsionaalsus .....	10
2.2.2 Kasutatavus .....	12
2.2.3 Usaldusväarsus .....	13
2.2.4 Jõudlus .....	14
2.2.5 Toetatavus .....	15
2.3 Andmebaasi arhitektuur .....	16
2.4 Tagasüsteem .....	18
2.4.1 Turvalisus .....	18
2.4.2 Suhtlus eesliidese ja andmebaasi vahel .....	19
2.5 Eesliides .....	20
2.5.1 Moodulid .....	20
2.5.2 Põhivaated .....	21
3. Tagasiside analüüs ja edasiarenduse plaanid .....	25
3.1 Tagasiside analüüs .....	25
3.2 Tuleviku arendusplaanid .....	29
Kokkuvõte .....	32
Viidatud kirjandus .....	34
Lisad .....	36
I. Valminud veebirakenduse lähtekood .....	36
II. Tagasiside saamiseks saadetud küsitlus .....	37
III. Tagasisideküsitluses kasutatud demovideo .....	39
Litsents .....	40

## Sissejuhatus

Praegu pole Tartu Ülikooli kursuse „Programmeerimine” (LTAT.03.001) aine jaoks olemas ühtset süsteemi, mis võimaldaks kursuse materjale hallata. Materjalid on kogutud peamiselt Google Drive’i ja Moodle’isse. Nende keskkondade abil on dokumente võimalik küll hoiustada, aga spetsiifiliste ülesannete või nende koostamis- ja hindamisjuhendite leidmine kujuneb keerukaks ja aeganõudvaks. Kuna üle aastate kasutusel olevad dokumendid on jaotatud eraldi Google Drive’i, Courseesi ja Moodle’i keskkondade vahel, siis on raske hinnata olemasolevate ülesannete kogust. Lisaks koguse hindamisele on keeruline aru saada, milliste teemade kohta on piisavalt ülesandeid olemas ning milliste jaoks oleks tarvis neid juurde teha. Seetõttu võib juhtuda, et uusi ülesandeid loodakse teemade kohta, mille jaoks on tegelikult juba piisavalt materjale olemas. Juhendite puhul võib juhtuda, et mõnda juhendit loodakse rohkem kui üks kord.

Selle bakalaureusetöö eesmärk oli koostada õppeainele „Programmeerimine” veebirakendus, mis võimaldaks kursuse dokumente hallata. Veebirakenduse arendamisel võeti eesmärgiks valmis teha süsteem, mis võimaldaks dokumentide muutmist, lisamist, filtreerimist ja allalaadimist. Lisaks võeti eesmärgiks üles seada arenduskeskkond, mida saab ka peale käesoleva bakalaureusetöö lõppu edasi kasutada.

Bakalaureusetöö on jagatud kolmeks osaks. Esimeses osas antakse ülevaade veebirakenduse olemusest ning arutatakse, millistest osadest see koosneb. Lisaks pannakse paika, millised tehnoloogiaid arendamiseks kasutama hakatakse. Teises osas kirjeldatakse loodava rakenduse nõudeid ja antakse ülevaade süsteemis olevate komponentide arhitektuurist ja nende ülesannetest. Kolmandas peatükis analüüsitakse kasutajate poolt antud tagasisidet süsteemile ning tehakse ülevaade edasistest arendusplaanidest. Selle bakalaureusetöö raames koostatud materjalid ja uuringu jaoks koostatud küsitlus on esitatud lisades.

# 1. Veebirakendus

Traditsiooniliselt jaotatakse veebirakendus kolmeks erinevaks osaks: eesliides, tagasüsteem ja andmetasand [1]. Eesliidest (ingl *frontend*) kasutatakse rakenduse kasutajale info kuvamiseks. Tagasüsteemis (ingl *backend*) toimub info töötlemine ja selle jagamine nii eesliidese kui andmetasandiga. Andmetasand on rakenduse andmebaas, kus hoitakse süsteemi jaoks vajalikke andmeid. Selline veebirakenduse ülesehitus jaotab kogu rakenduse osadeks, mis tähendab, et süsteemi arendamine muutub lihtsamaks, sest on kergem jälgida kogu info liikumist [1]. Nende osade realiseerimiseks on palju erinevaid viise tänu mitmetele raamistikele. Enne arendamise algust tuleks võimalusi kaaluda ja nende seast välja valida parimad raamistikud, mis püstitatud probleemi ära lahendaksid. Arendatav veebirakendus on mõeldud pikemaks kasutamiseks, mis tähendab, et on tähtis mõelda sellele, et peale esialgset arendust peaksid olemas olema ka edasiarenduse võimalused. Selle tagamiseks kasutatakse tihti versioonihaldussüsteeme.

## 1.1 Versioonihaldussüsteem

Versioonihaldussüsteemid on tarkvaraarenduses laialt levinud. Tänapäeval on suuremaid projekte ja rakendusi üsna keeruline ilma nende süsteemideta jätkusuutlikuks muuta, eriti kui projekti kallal töötab mitu inimest samal hetkel. Versioonihaldussüsteemid salvestavad iga faili puhul muudatusi ja muud kasulikku infot. Arendajatel on võimalik näha, kes failis parasjagu muudatusi on teinud ja millal nad seda tegid. See tuleb kasuks näiteks siis, kui on tekkinud mingi probleem ning on vaja lahenduse autorilt selgitusi küsida. Kuna versioonihaldussüsteemid salvestavad kogu faili ajalugu, siis see tähendab, et arendajatel on alati võimalus eelnev projekti seis taastada ja see avab arendajatele võimaluse katsetada riskantsemaid lahendusi [2]. Versioonihaldussüsteemid võimaldavad mitmel arendajal koostööd teha ning lihtsustavad uute arendajate tööd, kes hiljem projektiga liituvad. See tähendab, et lisades rakendusele versioonihaldussüsteemi, muutub rakenduse arendamine jätkusuutlikumaks. Käesoleva töö tegemiseks otsustati kasutusele võtta tarkvara Git<sup>1</sup>.

## 1.2 Populaarsed andmebaasid

Himanshu Kathuria on teinud ülevaate andmebaasidest ja nende populaarsusest aastal 2022. Oma uuringus ta arvestas Stack Overflow<sup>2</sup> poolt läbi viidud uuringut, milles osales umbes 74

---

<sup>1</sup> <https://git-scm.com/>

<sup>2</sup> <https://stackoverflow.com/>

000 inimest 181 erinevast riigist [3]. Lisaks Stack Overflow poolt tehtud küsitlusele arvestas Himanshu Kathuria ka teadmusbaasi DB-Engines<sup>3</sup> poolt antud edetabelit [4]. Kasutatud edetabelis olid haldussüsteemid reastatud nende populaarsuse järgi. Nende uuringute põhjal tuli esimesele kohale MySQL, mis on loodud Oracle'i<sup>4</sup> poolt. MySQLi populaarsus tuleb sellest, et tegemist on ühe vanima avalikult kättesaadavalt süsteemiga. Aja jooksul on tekkinud igasuguseid kasulikke funktsionaalsusi, mida on pidevalt optimeeritud ja edasi arendatud. MySQL on igapäevaselt kasutusel näiteks Facebooki, Uberi ja Youtube'i süsteemides. Populaarsuselt teine andmebaasihaldussüsteem on PostgreSQL. Süsteem avaldati aastal 1989 California ülikooli poolt. PostgreSQLi populaarsus tuleneb tema paindlikkusest. Erinevalt MySQList saab PostgreSQL hakkama erinevate JSON-tüüpi andmete haldamisega. PostgreSQLi on võimalik kasutada erinevates operatsioonisüsteemides nagu näiteks Windows, macOS, Linux ja Unix. PostgreSQLi süsteemid on igapäevaselt kasutusel Apple'i, Redditi, Nasa, Skype'i jt süsteemides [5]. PostgreSQLi paindlikkuse ja töö kirjutaja varsema kogemuse tõttu otsustati rakenduse andmetasandiks hakata kasutama PostgreSQLi.

### **1.3 Raamistikud**

Veebirakenduste arendamisel kasutatakse tihti erinevaid raamistikke. Raamistikud lihtsustavad eelkõige arendust, andes arendajatele võimaluse kasutada juba valmis ehitatud komponente, mida võib kohata tavapärares rakendustes. Nende komponentide olemasolu lihtsustab arenduskäiku ja annab rohkem aega teiste funktsionaalsuste teostamiseks [6].

Raamistik paneb paika vastava eesliidese või tagasüsteemi struktuuri. Struktuuri defineerimine on oluline selleks, et rakenduse ülesehitus järgiks kogu arenduse vältel ühtlast joont. Selline lähenemisviis tekitab vähem segadust. Lisaks, kui struktuur on paigas, siis on uute arendajate jaoks kergem rakendusega tutvuda, mis teeb rakenduse arendamise jätkusuutlikumaks [6].

#### **1.3.1 Angular**

Angular on populaarne eesliidese arendamiseks mõeldud raamistik, mis esmalt avaldati 2010. aastal Google poolt. Aastal 2023 ettevõtte SaM Solutions<sup>5</sup> poolt koostatud edetabelis, kus reastati enim kasutatud raamistikud eesliidese arendamiseks, on Angular saavutanud teise koha, mis näitab Angulari populaarsust veebirakenduste arendamises [7].

---

<sup>3</sup> <https://db-engines.com/en/>

<sup>4</sup> <https://www.oracle.com/>

<sup>5</sup> <https://www.sam-solutions.com/>

Angularil on mitmeid positiivseid külgi. Tegemist on ühe paindlikuma liidesega, mis hetkel saadaval on. Angulari abil on võimalik ehitada rakendusi kõikide veebibrauserite jaoks. Lisaks paindlikkusele tagatakse kiired laadimisajad käituseelse kompilaatori abil. Käituseelse kompileerimise korral kompileeritakse HTMLi ja TypeScripti kood JavaScriptiks veebilehe ehitamise ajal. See tähendab, et selleks ajaks, kui veebileht on valmis koodi alla laadima ja seda kasutama, on kood juba kompileeritud. Nii vähendatakse tunduvalt veebilehe kuvamise aega [8]. Angular kasutab kahesuunalist andmete sidumist, mis tähendab, et andmed vaadete ja koodi vahel on sünkroonis. Tänu sellele ei pea arendajad ise manuaalselt andmeid töötleva selleks, et olekud oleksid nii vaates kui koodis võrdväärased [9].

Angulari struktuur põhineb komponentidel. Komponentide abil defineeritakse rakenduse vaated. Lisaks vaadete defineerimisele on komponentidel võimalik kasutada erinevaid teenuseid. Teenustes on võimalik kirjutada loogikat, mis pole otseselt seotud vaate kuvamisega [10]. Kõige tavalisem funktsionaalsus, mida teenustes teostatakse, on eesliidese ja tagasüsteemi omavaheline suhtlus. Komponentidel põhinev struktuur hõlbustab koodi taaskasutamist. Taaskasutatava koodi kirjutamine säästab eelkõige aega. Juhul, kui koodi ei oleks võimalik taaskasutada, siis tuleks iga kord sama funktsionaalsus ümber kirjutada. Selle ümberkirjutamine aga vähendab aega, mida saaks kasutada uute ja originaalsete funktsionaalsuste arendamiseks. Lisaks aja säästmisele vähendab taaskasutamine vigade sattumist süsteemi. Tavaliselt on taaskasutatav komponent juba mõnda aega kasutusel olnud, mis tähendab, et seda on põhjalikult testitud. Selle ümberkirjutamine võib kaasa tuua vigast koodi [11].

Eesliidese arendamiseks otsustati valida Angular, sest töö autor on varsemalt seda kasutanud ja tunneb seda paremini kui edetabelis esimesel kohal olevat Reacti. Tänapäevaks on Angularil üks suurimaid kogukondi [12], mis teeb probleemidele lahenduste leidmise lihtsaks.

### **1.3.2 Spring Boot**

Vaadates edetabelit, kus tagasüsteemi raamistikud on järjestatud GitHubi repositooriumite arvu järgi, on näha, et Javale mõeldud raamistik Spring Boot on kaks korda rohkem kasutusel kui seda on tabeli teine koht Django [13]. Spring Booti populaarsus tuleneb sellest, et lisaks Javale toetab Spring Boot ka Kotlinit, Groovy ja Scalat.

Nagu kõik head raamistikud paneb projekti struktuuri paika Spring Boot ise. Raamistikule on omane hierarhiline struktuur. See moodustatakse nelja kihi abil ja iga kiht saab ainult enda alloleva või ülevaloleva kihiga suhelda. Need neli kihti on: esitluskiht, rakenduskiht, püsivuskiht ja andmebaasi kiht (vt joonis 1). Esitluskihi ülesanne on vastu võtta eesliidese poolt saadetud päringud ning töödelda päringu sees olevad JSON-andmed sobivale kujule, nii et need saaks edasi anda rakenduskihile. Rakenduskiht võtab saadud andmed vastu ning töötleb neid vastavalt vajadusele. Lisaks andmete töötlemisele hoolitseb rakenduskiht kasutaja, kes päringu saatis, autentimise eest. Püsivuskiht sisaldab endas loogikat, mis suudab teisendada Java objektid andmebaasi elementideks. Andmebaasi kiht võimaldab kasutada andmebaasi operatsioone, milleks on andmete salvestamine, kustutamine, muutmine ja pärimine [14].



Joonis 1. Spring Booti neli kihti.

Spring Booti üks eelis teiste raamistike ees seisneb selles, et see võimaldab arendajatel kasutada mitmesuguseid mooduleid [13]. Saadaval on kolm suuremat moodulit turvalisuse, andmete ja vahemälu töötlemiseks. Turvalisuse moodul võimaldab hõlpsasti teostada autentimist ning lisaks tagab ka elementaarsed turvameetmed levinud rünnakute vastu [15]. Andmete jaoks mõeldud moodul hoolitseb igasuguste andmetaseme keerukuste eest ja muudab andmetega töötamise lihtsamaks. Moodul suudab käsitleda relatsioonilise andmebaasi relatsioone ja eemaldab suures koguses korduvat koodi [16]. Vahemälu moodul võimaldab kasutada vahemälu rakenduse jooksumiseks. Vahemälu kasutamine tõstab rakenduse kiirust, mis omakorda annab positiivse kasutajakogemuse [17].

Java on üks populaarsemaid keeli läbi aegade [18], mis tähendab, et kasutades Spring Booti on saadaval suur kogukond, kes oskab probleemide leidmisel abiks olla. Lisaks on palju õppematerjale, mis vähendavad rakenduse arendamise keerukust. Kõikide eeltoodud omaduste pärast otsustati tagasüsteemi arendamiseks kasutada Spring Booti raamistikku.



## 2. Rakendus

Selles peatükis antakse ülevaade rakenduse nõuetest. Nõuded koostati kasutades mudelit, mis keskendub viiele osale. Nendeks osadeks on funktsionaalsus (ingl *functionality*), kasutatavus (ingl *usability*), usaldatavus (ingl *reliability*), jõudlus (ingl *performance*), ja toetatavus (ingl *supportability*). Edaspidi viidatakse mudelile kasutades akronüümi FURPS. Lisaks nõuete analüüsile kirjeldatakse täpsemalt andmebaasi andmete struktuuri ja relatsioone. Ühtlasi selgitatakse ka tagasüsteemi rolli rakenduses. Viimasena antakse ülevaade eesliidese arhitektuurist ja põhivaadetest. Nii tagasüsteemi kui ka eesliidese koodi kirjutamisel võeti arvesse Robert C. Martini raamatus olevaid nõuandeid, mis keskenduvad puhta ja arusaadava koodi kirjutamisele [19].

### 2.1 Nõuded

Altexsoft<sup>6</sup> on jaotanud rakenduse nõuded nelja erinevasse kategooriasse. On olemas ärilised, lahenduse, ülemineku ja kasutaja ehk osapoolte nõuded [20]. Käesoleva bakalaureusetöö eesmärki arvestades on kõige olulisemad lahenduse nõuded.

Altexsoft jagab lahenduse nõuded omakorda kaheks alamkategooriaks, milleks on mittefunktsionaalsed ja funktsionaalsed nõuded. Mittefunktsionaalsed nõuded defineerivad süsteemi üldomadused. Funktsionaalsed nõuded aga määravad rakenduse ülesanded ja funktsionaalsused [20]. Arendatava rakenduse nõuded said kirja pandud kasutades tuntud arvutitarkvarafirma Hewlett Packardi<sup>7</sup> poolt loodud mudelit nimega FURPS [21].

### 2.2 FURPS tehnika

FURPS on tarkvaraarenduses laialt levinud nõuete kirjapanemiseks mõeldud tehnika. Selle abil määratakse ära rakenduses vajaminevad nõuded viie erineva kriteeriumi abil.

FURPS tehnika viis kriteeriumit [21]:

1. funktsionaalsus;
2. kasutatavus;
3. usaldatavus;
4. jõudlus;

---

<sup>6</sup> <https://www.altexsoft.com/>

<sup>7</sup> <https://www.hpe.com/us/en/home.html>

## 5. toetatavus.

Funktsionaalsed nõuded panevad paika rakenduse ülesanded ja muud funktsionaalsused. Nende kirjapanemine annab arendajale hea ülevaate kõikidest vajaminevatest rakenduse elementidest. Hea ülevaade võimaldab paremini tööd planeerida ja välistab võimaluse, et töö lõpuks on midagi kahe silma vahele jäänud. Kasutatavuse alla langevad nõuded puudutavad kasutajakogemust ja rakenduse kasutamise meeldivust. Kolmandaks kriteeriumiks on usaldusväärsus. Usaldusväärsuse nõuded aitavad fikseerida süsteemi vastupidavuse ja tõrkekindluse. Jõudluse nõuded pööravad tähelepanu kogu süsteemi kiirusele ja võimekusele. Mõned nõuded jõudluse jaoks on näiteks maksimaalne aktiivsete kasutajate arv süsteemis, mälukasutus, päringule vastuse saamise kiirus ja vaadete kuvamise kiirused jne. Toetatavus aitab paika panna süsteemi hooldatavuse taseme. Lisaks kuulub selle alla ka rakenduse lahenduste muutmise või täiendamise hõlpsus.

### 2.2.1 Funktsionaalsus

Jaotis määrab ära rakenduse hädavajalikud funktsionaalsused. Lisaks seletatakse lahti ka funktsionaalsuste vajadus ja olulisus. Funktsionaalsuste nõuded on järgmised:

#### 1. Kasutaja peab saama lisada andmebaasi uusi dokumente.

Uute failide üleslaadimise funktsioon on vajalik selleks, et süsteem oleks kasutatav ka järgmiste aastate jooksul. Kui süsteem oleks mõeldud ainult käesoleva bakalaureusetöö tegemise ajaks koostatud materjalide haldamisega, siis salvestamise funktsionaalsus ei oleks hädavajalik. Sellisel juhul saaks olemasolevad materjalid manuaalselt andmebaasi sisestada. See tähendaks aga seda, et uute dokumentide salvestamise protsess oleks keerulisem ja saadaval ainult valitud inimeste jaoks. Üleslaadimise funktsiooni lisamine eesliidesesse toob salvestamise kättesaadavaks igale kasutajale ja lihtsustab uute dokumentide lisamise protsessi.

#### 2. Kasutaja peab saama lisada ülesannete dokumentidele näidislahendusi ja automaatkontrolle.

Ülesande jaoks on olulised näidislahendused ja automaatkontrollid. Moodle'is on näha, et kursuse „Programmeerimine” puhul on iga koduülesande puhul lisatud näidislahendused ning esituste õigsuse kontrollimiseks kasutatakse automaatkontrolle. Lisaks koduülesannetele on mõlemad variandid kasutusel mõningatel praktikumiülesannetel. Nii kontrollitööde kui ka

eksamite ülesannete jaoks on olemas näidislahendused. Vaadates, mis viisil dokumente kursusel kasutatakse, on selge, et lisaks tekstile on ülesande jaoks olulised ka näidislahendused ja automaatkontrollid.

### 3. Andmebaasi salvestatavad failid peavad olema esitatud .docx, .py või .sh vormingus.

Analüüsides eelmiste aastate jooksul kogutud materjale, on tegemist enamasti failidega, mida saab esitada mugavalt .docx failina. Kontrolltööde, praktikumite, eksamite ja lisaülesannete ülesanded on vormistatud nii, et neid saab esitada .docx vormingus. Lisaks ülesannetele on paljud juhendid koostatud nii, et neid saab esitada .docx failina. Kuna .docx fail sobib kõige paremini materjalide esitamiseks, on selle kasutamise võimalus süsteemis hädavajalik. Kõik näidislahendused on kursusel esitatud Python programmeerimiskeele lähtekoodifaili tüübina. Samuti on mõned automaatkontrollid .py vormingus, kuid leidub programme, mis on kasutusel .sh failina. Selleks, et ülesannete jaoks salvestada näidislahendusi ja automaatkontrolle, on tarvis kasutajatel võimaldada salvestada .py ja .sh tüüpi faile.

### 4. Kasutaja peab saama andmebaasist faile eemaldada.

Failide eemaldamise funktsionaalsus muudab süsteemi kasutajate jaoks kindlamaks ja kasutajasõbralikumaks. Juhul, kui kasutaja peaks üleslaadimisel protsessi käigus valima vale faili ja selle süsteemi salvestama, siis on tal võimalus viga parandada. Ebavajalike failide eemaldamise funktsionaalsus aitab hoida rakenduse sisu asjakohasena.

### 5. Kasutaja peab saama muuta andmebaasis olevaid faile.

Olemasolevate failide muutmise võimalus teeb süsteemi kasutajate jaoks mugavamaks. Pärast uue faili üleslaadimist võib kasutaja avastada, et faili sisu on ebakorrektne ja vajaks muutmist. Ilma selle funktsionaalsuseta tuleks kasutajal esmalt vigane fail süsteemist eemaldada, mõnes teises rakenduses parandada ja seejärel uuesti läbida üleslaadimise protsess. Sisu muutmise funktsionaalsus võimaldaks kasutajal ilma lisategevusteta faili sisu korrastada, mis säästaks kasutajate aega.

### 6. Rakendus peab võimaldama kasutajal dokumente filtreerida.

Vaadates eelmiste aastate materjalide hulka on selge, et kui süsteem kuvaks neid kõiki korraga, oleks soovitud dokumenti raske üles leida. Õige dokumendi leidmist saab lihtsustada erinevate

filtrite abil. Kuna rakendus on mõeldud hoiustama erinevate aastate faile, siis peab üheks filtriks olema aasta valimine. Lisaks aasta kaudu filtreerimisele peab süsteem võimaldama lisada dokumentide külge erinevaid märksõnu. Nende märksõnade kaudu peab rakendus võimaldama dokumente otsida. Märksõnadeks võivad olla näiteks erinevad teemad, mida ülesanne kontrollib või nädalad, millesse dokument kuulub. Lisaks peab süsteem võimaldama dokumente filtreerida pealkirjade järgi.

#### 7. Rakendus peab pakkuma elementaarseid turvameetmeid.

Rakendus sisaldab endas Tartu Ülikooli kursuse materjale, mis tähendab, et süsteem peab olema turvatud. Vastasel juhul võivad rakenduses olevad materjalid kättesaadavad olla kursusel osalevatele tudengitele, kes saaksid neid pahameelselt ära kasutada, et oma tulemusi ebaausalt tõsta. Süsteemis peab olema nii tavakasutaja kui ka administraatori kasutaja. Tavakasutajatel on võimalik ennast süsteemi registreerida ning pärast registreerimist peab olema administraatoril võimalik kasutaja aktiveerida. Aktiveerimata kasutaja ei tohi süsteemi sisule ligi pääseda. Lisaks aktiveerimisele peab olema administraatoril võimalik kasutajaid süsteemist eemaldada.

### 2.2.2 Kasutatavus

Jaotis määrab ära süsteemi kasutatavust mõjutavad nõuded. Nõuded pööravad tähelepanu kasutajakogemuse parandamisele. Kasutatavuse kohta käivad nõuded on järgmised:

#### 1. Süsteemis peab olema komponentide vaheline navigeerimine intuitiivne.

Selgesti arusaadav navigeerimissüsteem teeb rakenduse kasutamise kergesti mõistetavaks ja vähendab kasutajate frustratsiooni. Loogiline navigeerimine vähendab dokumentide otsimise aega ja seetõttu soodustab kasutajakogemust. Kasutajakogemuse parandamine tähendab omakorda seda, et kasutaja jääb rakendusega rahule ja külastab veebilehte ka tulevikus. Rakendus peab kasutama standardkohast horisontaalset navigeerimisriba, mis tähendab, et kasutaja jaoks on igas vaates kõik grupid nähtaval, mis hõlbustaks nende vahel navigeerimist.

#### 2. Süsteem peab olema kasutatav ka vaegnägemisprobleemide all kannatavate inimeste poolt.

Maaailma Terviseorganisatsiooni andmetel kannatavad maailmas vähemalt 2,2 miljardit lühivi või kaugnägelikkuse seotud probleemide all [22]. Lisaks organisatsiooni Clinton Eye

Associates<sup>8</sup> andmetel on maailmas ligi 300 miljonit inimest, kes kannatavad värvipimeduse all [23]. Selleks, et rakenduse vaated rohkem kasutajasõbralikud oleksid, tuleb esmalt valida värvid, mis on kergesti eristatavad ja sobiva kontrastsusega. Kasutaja jaoks ei tohi teated, ikoonid ja muud sümbolid sõltuda ainult värvusest. Rakenduses olevatel elementidel peab olema juures selgitavaid tekste. Lisaks värvidele peab vaadetes olev tekst olema piisava suurusega.

### 3. Kasutajad peavad saama tegevuste kohta teadete kaudu tagasisidet.

Andes kasutajatele tagasisidet muutub rakenduse kasutamine meeldivamaks. Pärast tegevuse sooritamist soovib kasutaja saada teadet, et tema tegevus sai registreeritud ning töödeldud süsteemi poolt. Vastasel juhul võib kasutaja segadusse jääda ja muutuda ebakindlaks. Ilma tagasisidet saamata ei saa kasutaja kindel olla, et dokumendi salvestamine või üleslaadimine oli edukas. Seetõttu võib süsteemi rikke korral kasutaja poolt tehtud töö kaotsi minna, ilma et kasutaja sellest teadlik oleks. Lisaks on tänu teadete kuvamisele kasutajatel võimalik raporteerida süsteemi kasutamise ajal leitud vigu. Nende raportite põhjal on arendajal võimalik rakendust täiustada ja süsteemi meeldivamaks muuta.

#### 2.2.3 Usaldusväärsus

Jaotis defineerib ja seletab lahti nõuded, mis puudutavad rakenduse kasutamist normaaltingimustes. Lisaks seletatakse ja põhjendatakse nõuete olulisust süsteemile. Usaldusväärssust puudutavad nõuded on järgnevad:

#### 1. Rakendus peab sisaldama võimalikult vähe tõrkeid.

Kindlustades, et rakendus ei lõpeta oma tööd tõrke tõttu, tõstetakse süsteemi kvaliteeti ja kasutajakogemust. Kui rakendus ootamatult lõpetaks oma töö, siis võib juhtuda, et osad süsteemis olevad dokumendid lähevad kaotsi või saavad muul moel kahjustada. Lisaks andmete kahjustamisele muutub rakendus kasutajate jaoks ebameeldivamaks ja viiks frustratsioonini. Kui kasutajad on rahulolematud, siis nad ei kasuta süsteemi ning rakendus kaotaks oma eesmärgi.

---

<sup>8</sup> <https://clintoneye.com/>

## 2. Rakendus peab toime tulema tõrgete töötlemisega.

Rakendus ei tohi lõpetada oma tööd tõrgete korral ja peab tagama, et salvestatavad andmed oleksid nõuetele vastavad. Kui juhtub, et kasutaja esitab üleslaadimisel vales vormingus dokumendi, siis peab süsteem selle tagasi lükkama ja kasutajat sellest teavitama. Rakendus ei tohi oma tööd lõpetada ebakorrekse sisendi tõttu. Näiteks, kui kasutaja esitab vales vormingus faili, siis rakendus ei tohi oma tööd lõpetada, vaid peab võimaldama kasutajal jätkata rakenduse kasutamist.

## 3. Rakendus peab olema kasutatav laialt levinud brauseritega.

Similarwebi<sup>9</sup> poolt pakutavate andmete põhjal on kolm kõige levinumat veebibrauserit Chrome, Safari ja Edge. Chrome on teistega võrreldes tunduvalt populaarsem [24]. Chrome populaarsusest sõltumata tuleb arvestada ka teiste võimalustega, sest iga kasutaja on erinev. Erinevate brauserite kasutamisevõimaluse tagamiseks tuleb arvestada eesliidese arendamise juures levinumaid standardeid, mis on loodud Tim Berners-Lee poolt [25]. Standardite järgi peab süsteem kasutama HTML-, CSS- ja HTTP-protokolli. Nende kasutamine tagab selle, et rakendus jääb kasutuskõlblikuks mitmes veebibrauseris.

### 2.2.4 Jõudlus

Jaotis määrab ja seletab ära rakenduse nõuded, mis puudutavad süsteemi kiirust. Nõuded panevad paika maksimaalsed laadimise ja päringutele vastuste andmise ajad. Lisaks määratakse ära, kui palju kasutajaid võib samaaegselt süsteemi kasutada. Jõudlust käsitlevad nõuded on järgnevad:

#### 1. Vaadete laadimine ei tohi võtta kauem kui neli sekundit.

Digitaalse turundusagentuuri Portenti<sup>10</sup>, koostatud uuringu põhjal loetakse veebilehte tõhusaks, kui andmed suudetakse laadida kasutaja jaoks nelja sekundi jooksul. Veebilehe konversioonimäär aitab hinnata, kui suur osa kasutajatest täidab soovitud tegevuse. Kõrge konversioonimäär tähendab, et rakendus suudab kasutajaid veenda ja juhtida tegema nende soovitud tegevusi. Madal konversioonimäär näitab seda, et rakendus ei täida oma ülesannet ja kasutajatel pole võimalik soovitud tegevusi kergelt teostada. Uuringu abil tehti kindlaks, et kui veebileht võtab enam kui viis sekundit laadimiseks, siis väheneb rakenduse konversioonimäär

---

<sup>9</sup> <https://www.similarweb.com/>

<sup>10</sup> <https://www.portent.com/>

vähemalt poole võrra. Lisaks konversioonimäära tõstmisele tõstavad kiired laadimisajad kogu rakenduse kvaliteeti ja kasutajakogemust [26].

2. Päringutele vastuse saamine ei tohi võtta kauem kui üks sekund.

Eesliidese poolt tehtud päringutele peaks vastuse saama keskmiselt 200 millisekundi jooksul. Selline kiirus eemaldab kasutaja jaoks näiliselt ooteaja ja jätab mulje, et vastus tegevusele saabus kohe. Vastuse saamise ülemiseks piiriks loetakse ühte sekundit. Ühesekundilise ooteaja jooksul ei pane ilmselt kasutaja veel tähele, et tegevuse järel tuleb oodata. Kõikide päringute, mis võtavad kauem kui üks sekund, jaoks tuleb leida lahendus. Vastasel korral hakkavad need päringud mõjutama kasutajate rahulolu ja süsteemi kvaliteeti [27].

3. Süsteem peab olema korraga kasutatav vähemalt viieteistkümne kasutaja poolt.

Käesoleva bakalaureusetöö ajal on rakenduse kasutamine enamasti mõeldud ainult isikutele, kes on kursuse „Programmeerimine” õppejõudude ja õppeassistentide nimekirjas. Tavaliselt on see nimekiri piirdunud umbes viieteistkümne inimesega kursuse toimumise ajal, mis tähendab, et süsteem peab võimaldama nendel isikutel samaaegselt süsteemi ilma tõrgeteta kasutada. Tavaliselt suudavad hästi ehitatud veebilehed toime tulla ligi kahesaja viiekümne kasutajaga, aga antud bakalaureusetöö skoobis pole see vajalik. Süsteem peab aga võimaldama tulevikus seda arvu tõsta.

### **2.2.5 Toetatavus**

Järgnev jaotis annab ülevaate nõuetest, mis mõjutavad rakenduse toetatavust. Defineeritakse ja põhjendatakse ära nõuded, mis peavad rakenduse arendamise tegema jätkusuutlikuks. Toetatavust puudutavad nõuded on järgnevad:

1. Rakendus peab arenduseks kasutama jätkusuutlikke tehnoloogiaid.

Jätkusuutlike tehnoloogiate kasutamine tähendab, et rakendus suudab areneda. Kuna veebirakenduste arendamine on tarkvara loomises tähtsal kohal, siis sellega seonduvad tehnoloogiad saavad pidevalt uuendusi, mis tõstavad süsteemide kvaliteeti. Seetõttu peab arendatav süsteem olema ehitatud kasutades tuntuid ja populaarseid tehnoloogiaid. Angular ja Spring Boot on tänapäeval populaarsed ja suure kogukonnaga ning rahuldavad seda nõuet.

2. Süsteem peab olema ka arendatav peale käesoleva bakalaureusetöö lõppu.

Kuna bakalaureusetöö jaoks on antud piiratud aeg, siis ei ole võimalik rakendust lõpuni välja arendada. See tähendab, et tuleb säilitada arenduskeskkond nii, et süsteem oleks kättesaadav peale töö lõppu. Arenduskeskkonna säilitamine on tarvis aja jooksul tekkinud vigade ja tõrgete parandamiseks. Lisaks on võimalik uutel arendajatel samuti oma panus anda. Bakalaureusetöö tegemise hetkel kasutatakse koodibaasi säilitamiseks repositooriumi, mis on loodud keskkonda GitHub<sup>11</sup>. Repositoorium sisaldab endas ka põhjaliku juhendit, kuidas rakendust lokaalseks arendamiseks üles seada.

### **2.3 Andmebaasi arhitektuur**

Peatükk tutvustab andmebaasi salvestatavaid andmeid. Lisaks antakse ülevaade andmete relatsioonidest.

Rakenduse põhilisteks andmeteks on failid. Käesoleva töö raames ehitatakse struktuur üles kahe dokumenditüübi peale, milleks on ülesanded ja juhendid. Kuna ülesannete koostamis- ja hindamisjuhendid on loodud nii, et neid saab kasutada üldisemalt, siis on võimalik nende käsitlemine ülesannetest eraldiseisvaks teha. Lisaks dokumendi enda sisule võimaldab süsteem salvestada dokumentide jaoks metaandmeid ja lisafaile. Metaandmete alla kuuluvad õppeaasta ja kursus. Ülesannete puhul on võimalik lisada ka märksõnu ja lisafaile nagu näidislahendused ja automaatkontrollid.

Käesoleva bakalaureusetöö tegemise raames on dokumentide jaoks määratud kaks ühist tunnust. Nendeks on aasta, mis iseloomustab, millal dokumenti kasutati või kasutatakse. Teiseks tunnuseks on kursuse kood, mis näitab, mis kursuse jaoks dokument mõeldud on. Nende salvestamine on kasutajate jaoks oluline, sest nende abil on kergem dokumente üles leida.

Juhendite tüübi jaoks rohkem tunnuseid ei salvestata. Ülesannete jaoks on lisatud aga veel kolm tunnust. Kasutajatel on võimalik ülesande faili jaoks defineerida märksõnu, mis iseloomustavad ülesande olemust ja sisu. Näiteks ülesande jaoks, mis kontrollib tsükli, järjendite ja tingimuslausete oskusi, oleksid sobivad märksõnad „tsükkel”, „järjend” ja „tingimuslause”. Nende abil on hiljem kasutajatel kergem üles leida ülesandeid, mis sisaldavad kindla märksõna

---

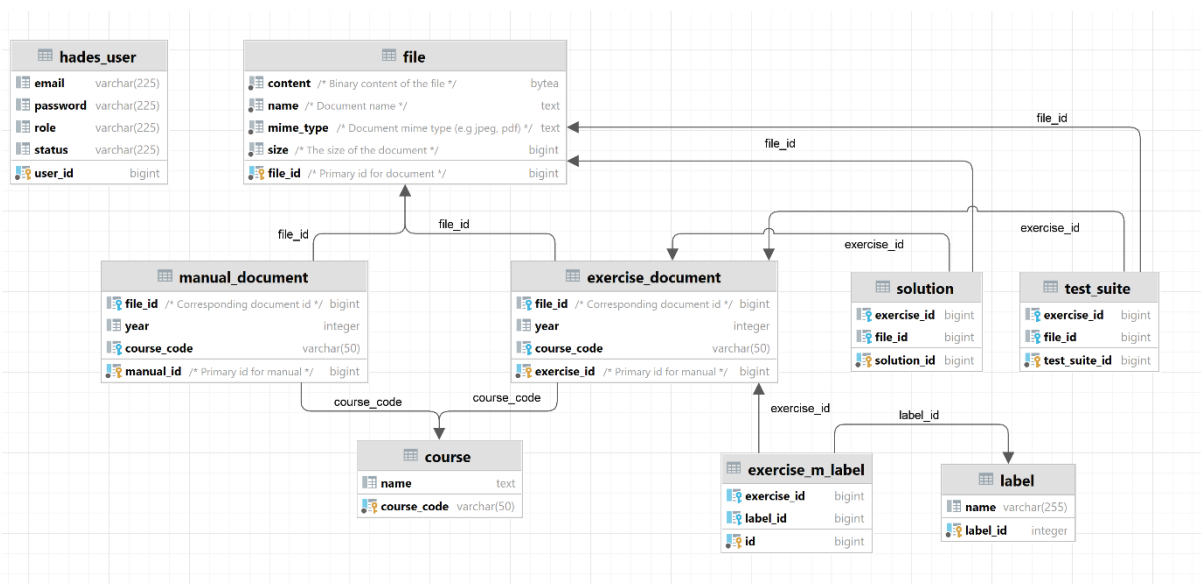
<sup>11</sup> <https://github.com/KaurTaal/hades>



kasutamist. Kuna ülesandel saab olla mitu märksõna ja üks märksõna saab olla mitmel ülesandel, siis on tegemist mitu-mitmele seosega (vt joonis 2).

Ülesande jaoks saab lisada näidislahendusi. Näidislahendused on olulised selleks, et hiljem hindajatel oleks kergem kontrollida esitatud lahenduse õigsust. Lisaks hindamiste kergendamiseks on kursustel kombeks näidislahendusi tudengite jaoks avalikustada, et tudengid saaksid nendega oma lahendusi võrrelda. Kuna nõueteanalüüsis sai kindlaks tehtud, et näidislahendused on üks oluline osa ülesandest, siis võimaldab ka süsteem neid lisada. Ülesandel saab olla mitu näidislahendust, aga üks näidislahendus saab olla ainult ühel ülesandel. See tähendab, et tegemist on üks-mitmele seosega (vt joonis 2).

Viimaseks lisatunnuseks ülesande dokumendi jaoks on automaatkontrollid. Automaatkontrolle kasutatakse hindamisprotsessi kiirendamiseks. Nende abil automatiseeritakse lahenduse õigsuse kontrollimine ja see vähendab märgatavalt hindamiseks kuluvat aega. Automaatkontroll on oluline osa ülesandest, mistõttu võimaldab süsteem ülesande jaoks salvestada ka automaatkontrolli faile. Reeglina on ülesandel mitu automaatkontrolli ja üks automaatkontroll vastab kindlale ülesandele. See tähendab, et sarnaselt näidislahendusega on tegemist üks-mitmele seosega (vt joonis 2).



Joonis 2. Andmebaasi struktuur.

Selline andmemudel tagab kasutajate jaoks kõik nõueteanalüüsis koostatud vajadused. Lisaks failidele hoitakse andmebaasis ka süsteemi kasutajaid (vt joonis 2). Kuna töö tegemise hetkel pole otsest seost kasutaja ja dokumentide vahel, siis nende vahelist seost andmebaasi ei teki

ning kasutajate tabel jääb teistest tabelitest eraldatuks. Kasutajate puhul salvestatakse e-mail, krüpteeritud parool, roll ja staatus.

## **2.4 Tagasüsteem**

Järgnev peatükk annab ülevaate tagasüsteemist. Käsitletakse nii turvalisust kui ka eesliidese ja andmebaasi suhtluse delegeerimist. Analüüsitakse väärtusi, mida toob tagasüsteemile Spring Security. Viimasena tehakse täpsem ülevaade tagasüsteemi arhitektuurist ja tema ülesannetest.

### **2.4.1 Turvalisus**

Süsteemis hakatakse hoidma Tartu Ülikooli kursuseid puudutavaid andmeid, mille abil hinnatakse tudengite oskusi. See tähendab, et salvestatavad andmed on tundlikud ning süsteem peab omama elementaarseid turvanõudeid. Järgnev peatükk vaatleb täpsemalt, mis võimalusi Spring Security liides pakub, et rakenduse tagasüsteem turvaliseks muuta. Lisaks seletatakse kasutajate liike ning nende iseärasusi.

Kogu turvalisuse eest hoolitseb tagasüsteemi Springi liides Spring Security. Spring Security võimaldab rakenduse jaoks paigaldada paindliku autentimiskihi. Liides võimaldab enamasti käsitleda kahte põhilist aspekti, mis puudutab rakenduse turvamist. Nendeks on autentimine ja autoriseerimine. Lisaks nendele pakub Spring Security ka vaikimisi kaitset laialtlevinud rünnakute vastu [28].

Autentimist kasutatakse isiku kindlaks tegemiseks. Kui tagasüsteem saab eesliidese poolt päringu, siis enne andmete avalikustamist tehakse kindlaks, kas tegemist on isikuga, kellel on süsteemile ligipääs. Arendatav süsteem kasutab kasutajate autentimiseks nende poolt sisestatud e-maili ja parooli. Pärast seda, kui isik on kindlaks tehtud, minnakse edasi autoriseerimise juurde. Autoriseerimise etapis tehakse kindlaks, kas isikul on piisavalt kõrge ligipääsu tase, et päritavatele andmetele ligi pääseda. Käesoleva bakalaureusetöö raames on süsteemile lisatud kahte liiki kasutajaid. Üks nendest on administraator ja teine tavakasutaja. Erinevalt tavakasutajast on administraatoril ligipääs ka kasutajate haldamisele. Kui kasutaja ennast süsteemi registreerib, siis administraator peab registreeritava kasutaja esmalt aktiveerima, enne kui kasutaja süsteemile ligi pääseb. Nii tehakse kindlaks, et iga registreerinud kasutaja ei saa dokumentidele ligi. Spring Security pakub eri tasemetel autoriseerimist. Täpsemalt pakub liides nii päringutasemelist kui ka spetsiifilise meetodi tasemelist autoriseerimist. Kuna rakendusel

on töö tegemise hetkel ainult kahte liiki kasutajaid, siis piirdub süsteem päringutaseme autoriseerimisega.

## **2.4.2 Suhtlus eesliidese ja andmebaasi vahel**

Pärast kasutaja autentimist hakkab tagasüsteem juhtima andmete liiklust eesliidese ja andmebaasi vahel. Järgnev peatükk annab ülevaate tagasüsteemi toimingutest, mis tagavad, et andmebaasist tulevad andmed oleksid eesliidese jaoks sobival kujul. Lisaks hoolitseb tagasüsteem ka selle eest, et eesliidese poolt saadetud andmed saaksid andmebaasi salvestatud.

Tagasüsteemil on neli erinevat kihti. Iga kiht koosneb seitsmest moodulist. Tagasüsteemi seitse moodulit on:

1. failimoodul;
2. ülesandemoodul;
3. juhendimoodul;
4. kasutajamoodul;
5. autentimisemoodul;
6. märksõnamoodul;
7. kursusemoodul.

Mooduli alla võib kuuluda erinevaid komponente olenevalt sellest, mis tüüpi mooduliga täpsemalt tegemist on. Enamasti koosneb moodul neljast komponendist. Esimeseks komponendiks on kontrolleri, mis vastutab eesliidese poolt saadetud päringute vastuvõtmise eest. Lisaks andmete vastuvõtmisele teisendab kontrolleri vastuvõetud andmed teenuste jaoks sobivateks andmetüüpideks. Kontrolleriist liiguvad andmed edasi teise komponendi juurde, milleks on teenus. Teenus on jaotatud kaheks osaks. Esimene osa on tavaline liides, mis defineerib teenuses saadaolevad meetodid ja paneb paika meetodite tagastustüübid. Liideseid kasutatakse ka koodi dokumenteerimiseks. Liideses olevate meetodite dokumenteerimine muudab koodi arusaadavamaks ja kergesti loetavamaks. Dokumentatsiooni abil on järgmistel arendajatel kergemini süsteemi tundma õppida. Liidese teine osa sisaldab liideses defineeritud meetodite teostust. Teenuse praktiline pool töötleb kontrolleri poolt saadud andmed andmebaasi jaoks sobivale kujule. Lisaks hoolitseb teenus selle eest, et eesliidesele saadetavad andmed oleksid sobivas vormingus, et neid kasutajale kuvada. Kolmas oluline osa moodulist on repositoorium. Repositoorium on põhiline viis andmebaasiga suhtlemiseks. Repositooriumid sisaldavad endas kõiki vajalikke elemente, et andmebaasi andmeid nii salvestada, muuta, pärida

kui ka kustutada. Neljandaks komponendiks on moodulispetsiifilised testid. Testide abil kontrollitakse teenuste sisu õigsust. Testid on olulised selleks, et rakenduse arendus oleks jätkusuutlik. Ilma testideta oleks olemasolevate meetodite muutmine riskantne. Testid aitavad kindlaks teha, et pärast muudatuste tegemist jätkab meetod oma tööd nii, nagu algselt mõeldud on. Lisaks põhikomponentidele on võimalik mooduli jaoks defineerida abistavaid klasse. Kõige tavapärasem abistav klass on utiliidiklassid. Utiliidiklassidesse lisatakse enamasti kood, mida saab kasutada üle kogu rakenduse. Kuna rakendusel on kolm moodulit, mis kasutavad faile, siis selleks, et vältida korduvat koodi, saab defineerida utiliidiklassi, mis sisaldab loogikat failide töötlemiseks. Läbi utiliidiklassi on võimalik koodi taaskasutada ning nii välditakse koodi dubleerimist.

## **2.5 Eesliides**

Peatükk tutvustab rakenduses olevat eesliidest. Esmalt täpsustatakse, millist arhitektuuri eesliides oma töö tegemiseks kasutab. Lõpuks antakse ülevaade eesliidese põhivaadetest.

### **2.5.1 Moodulid**

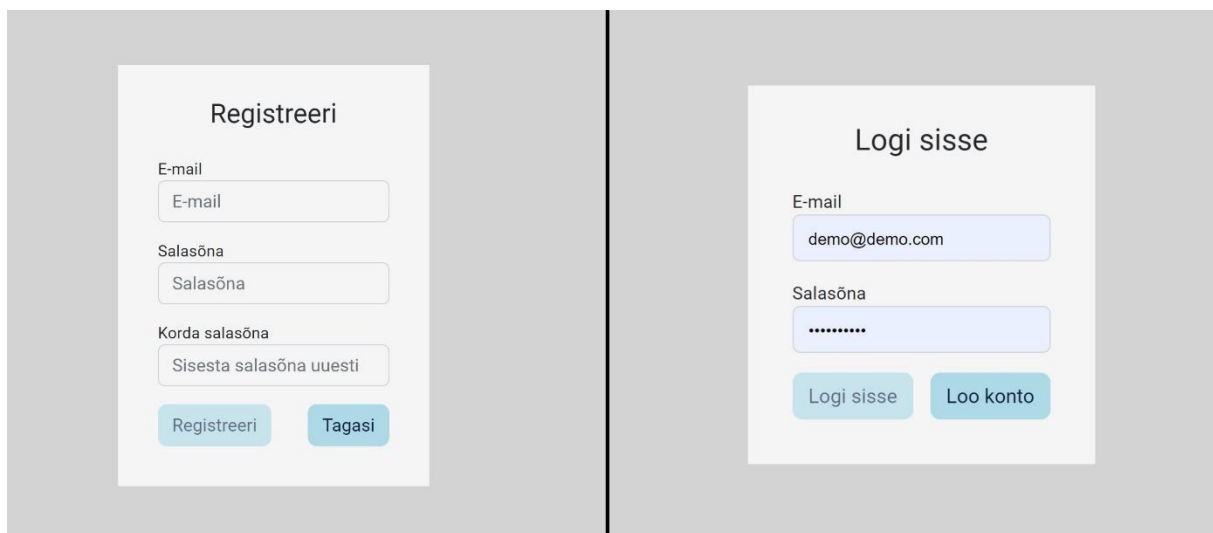
Sarnaselt tagasüsteemile on eesliidese arhitektuur üles ehitatud moodulite abil. Angulari rakenduste juures on võimalik kasutada Angulari spetsiifilist moodulitesüsteemi. Igas Angulari rakenduses defineeritakse algul üks peamoodul. Peamoodulisse on võimalik lisada alammoduleid, mis aitavad rakenduse laiali jaotada erinevateks valdkondadeks. Lisaks alammoduleite lisamisele on võimalik moodulites defineerida komponente, teenuseid ja muid abistavaid elemente. Kuna Angular lubab ühte elementi defineerida ainult ühes moodulis, siis on võimalik paika panna, millised elemendid moodul ekspordib. Eksporditud elemendid on nähtavad teistele moodulitele ja neid on võimalik moodulitesse importida [29].

Käesoleva töö raames arendatud eesliidese arhitektuuri saab paika panna kolme mooduli abil. Peamoodulis, mis on olemas igal Angulari rakendusel, defineeritakse kaks alammoodulit. Esimeseks alammooduliks on avalik moodul. Avalik moodul sisaldab komponente, mille abil on kasutajatel võimalik sisse logida või ennast registreerida. Samuti sisaldab moodul endas teenust, mille abil on võimalik suhelda tagasüsteemi autentimisemooduliga. Teiseks alammooduliks on kaitstud moodul. Selle kasutamiseks peab kasutaja ennast edukalt süsteemi sisse logima. Kaitstud moodul defineerib kõik rakenduse muud vaated, mida läheb vaja dokumentide ja kasutajate haldamiseks. Moodulite abil on võimalik eesliides väiksemateks osadeks teha, mis aitab paremini mõista, kuidas eesliides oma tööd teostab.

## 2.5.2 Põhivaated

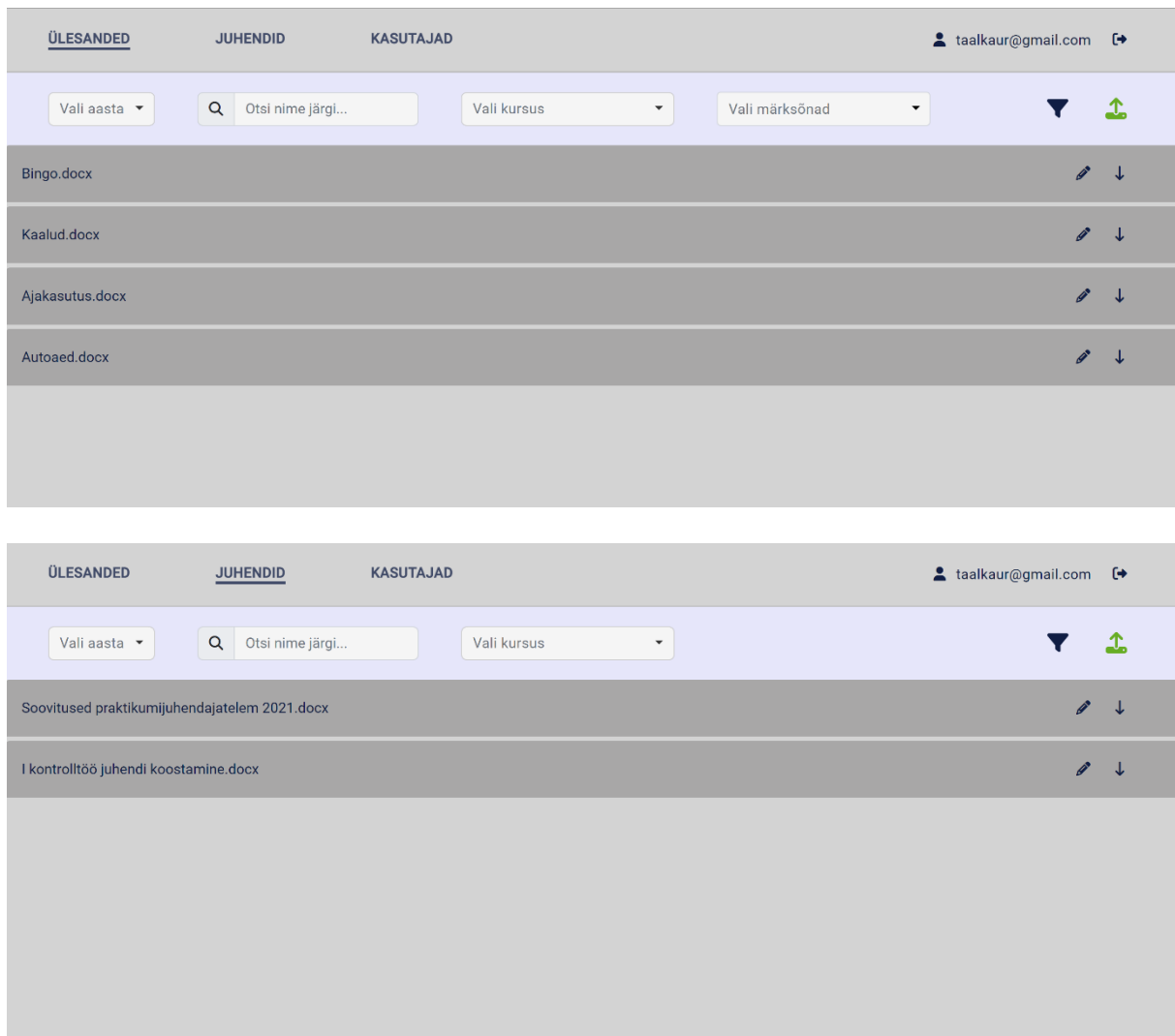
Jaotis teeb ülevaate rakenduse erinevatest vaadetest. Peatükk käsitleb vaateid kahes erinevas kategoorias. Esimene kategooria sisaldab avalikke vaateid ning teine privaatseid ehk kaitstud vaateid.

Avalik moodul koosneb kahest komponendist. Need komponendid koostavad sisselogimise ja registreerimise vaate (vt joonis 3). Konto loomisel tuleb kasutajal sisestada e-maili aadress, mis käitub süsteemi kasutajanimena. Lisaks e-maili sisestamisele tuleb kasutajal valida salasõna. Salasõna tuleb sisestada kaks korda ning mõlemad sisestused peavad omavahel ühtima. Pärast registreerimist saab kasutaja edasi minna sisselogimise vaatesse. Süsteemile ligipääsemiseks peab kasutaja sisestama oma e-maili ja salasõna.

The image shows two side-by-side web forms. The left form is titled 'Registreeri' (Register) and contains three input fields: 'E-mail' with the placeholder 'E-mail', 'Salasõna' (Password) with the placeholder 'Salasõna', and 'Korda salasõna' (Repeat password) with the placeholder 'Sisesta salasõna uuesti'. Below these fields are two buttons: 'Registreeri' and 'Tagasi' (Back). The right form is titled 'Logi sisse' (Log in) and contains two input fields: 'E-mail' with the placeholder 'demo@demo.com' and 'Salasõna' with the placeholder '.....'. Below these fields are two buttons: 'Logi sisse' and 'Loo konto' (Create account).

Joonis 3. Sisselogimise (vasakul) ja registreerimise (paremal) vaated.

Kaitstud moodul koosneb kolmest komponendist. Kuna süsteem võimaldab lisada kahte tüüpi dokumente, siis on mõlema kategooria jaoks eraldi vaated, sest nende eraldamine teeb dokumendi otsimise lihtsamaks. Esimene vaade on ülesannete ning teine juhendite jaoks (vt joonis 4). Vaated on omavahel sarnased ja erinevad ainult filtreerimisvalikute osas. Mõlemad vaated sisaldavad vastava kategooria allalaaditud dokumentide loetelu. Lisaks dokumentidele on nähtaval ka filtreerimise valikud. Kuna juhendite jaoks pole võimalik salvestada märksõnu, siis ei ole vastavas vaates dokumentide filtreerimist märksõnade kaudu võimalik teostada. Ülesannete vaates on aga kõik filtreerimisvalikud kasutaja jaoks saadaval. Lisaks dokumentide loetelule on vaadete ühiseks komponendiks veel faili üleslaadimiseks mõeldud nupp, mille abil on võimalik alustada uue faili lisamise protsessi.



Joonis 4. Ülesannete (üleval) ja juhendite (all) vaated.

Juhul, kui sisse logitud kasutaja on süsteemi administraator, on võimalik ligi pääseda kasutajate haldamiseks mõeldud vaatele (vt joonis 5). Kasutajate haldamise vaates kuvatakse tabelina kõik süsteemi kasutajad. Tabelis on näha kasutaja e-mail, roll ja staatus. Kasutajat on võimalik määrata administraatoriks või tavakasutajaks. Staatus määramise puhul on võimalik valida kahe valiku vahel. Kasutaja saab olla kas aktiveeritud või registreeritud. Registreeritud olek tähendab, et kasutaja on deaktiveeritud ja kasutajal pole võimalik kaitstud vaadetele ligi pääseda. Lisaks rolli ja staatus määramisele on võimalik kasutajat kustutada.

ÜLESANDED		JUHENDID	KASUTAJAD	taalkaur@gmail.com	
E-mail	Roll	Staat		Staat	
taalkaur@gmail.com	Administraator	Kasutajaks	Aktiveeritud	Deaktiveeri	
test@test.com	Administraator	Kasutajaks	Aktiveeritud	Deaktiveeri	

Joonis 5. Kasutajate haldamise vaade.

Kõikide vaadete juures on kasutatud selgitavaid tekste ja värvikombinatsioone, mis on kergesti eristatavad. Selline kujundus teeb süsteemi kasutajasõbralikumaks ja ei nõua kasutajatelt täiendavat keskendumist, et funktsionaalsusi üles leida. Kergesti eristatavad elemendid teevad süsteemi kättesaadavamaks vaegnägijatele. Kasutades tehnoloogiat Web Disability Simulator<sup>12</sup> on võimalik veebilehe väljanägemist vaadata eri nägemispuuete puhul. Kui simuleerida veebilehte kasutaja puhul, kes ei suuda hästi eristada punast ja rohelist värvust, on näha, et rakenduses olevad funktsionaalsused säilitavad oma selguse (vt joonis 6).

<sup>12</sup> <https://chrome.google.com/webstore/detail/web-disability-simulator/olioanlbgpmdlgjnnampnnlohigkja>

## Registreeri

E-mail

Salasõna

Korda salasõna

ÜLESANDED	JUHENDID	KASUTAJAD	taalkaur@gmail.com	
Vali aasta	Otsi nime järgi...	Vali kursus	Vali märksõnad	
Bingo.docx				
Kaalud.docx				
Ajakasutus.docx				
Autoaed.docx				

ÜLESANDED	JUHENDID	KASUTAJAD	taalkaur@gmail.com	
E-mail	Roll		Staatus	
taalkaur@gmail.com	Administraator	Kasutajaks	Aktiveeritud	Deaktiveeri
test@test.com	Administraator	Kasutajaks	Aktiveeritud	Deaktiveeri

Joonis 6. Rakenduse vaated läbi punase ja rohelse värvi eristamise raskuse puhul.

Kuna selgitavad tekstid annavad kasutajale piisavalt informatsiooni vastava nupu tegevusest, ei pea kasutajad lähtuma ainult ikoonidest või värvidest. Nii jääb rakendus kättesaadavaks ka inimestele, kellel on erinevaid nägemisega seotuid probleeme.



### 3. Tagasiside analüüs ja edasiarenduse plaanid

Loodud rakendusele tagasiside saamiseks koostati tagasisideküsitlus. Küsitluse abil hinnati valminud rakenduse vastavust eelnevalt kirjapandud nõuetele. Vastajate käest uuriti, mil viisil nad dokumente haldavad ja kas nad oleksid valmis uut süsteemi kasutama. Küsitluses paluti vastajatel hinnata süsteemis olevate funktsionaalsuste olulisust. Viimasena koguti vastajate käest erinevaid mõtteid, mida saaks tulevikuplaanide jaoks kasutada. Tagasisideküsitluse osalejateks olid kursuste „Programmeerimine”, „Programmeerimise alused” (MTAT.03.236), „Programmeerimise alused II” (MTAT.03.256) ja „Objektorienteeritud programmeerimine” (LTAT.03.003) õppejõud. Kokku vastas küsitlusele kümme inimest. Koostatud küsitlus on esitatud lisades. Lisaks tagasisideküsitluse analüüsile antakse ülevaade tuleviku arendusplaanidest.

#### 3.1 Tagasiside analüüs

Küsitlus sisaldas endas videot, mis tutvustas veebirakenduse põhilisi funktsionaalsusi. Video abil said vastajad rakendusega tutvuda ning seejärel anda tagasisidet kaheteistkümnele väitele. Väiteid sai hinnata viie palli skaalal, kus madalam hinnang sümboliseeris väitega mittenõustumist ja kõrgem hinnang väitega nõustumist. Skaala madalaim ja kõrgeim hinnang olid vastavalt 1 ja 5. Lisaks väite hinnangule oli vastajatel võimalik lisada täpsustavaid kommentaare. Väited koostati kasutades eelnevalt püstitatud nõudeid. Väidetele antud hinnangute abil hinnatakse loodud rakenduse vastavust püstitatud nõuetele.

Väited on kokku võetud tabelis 1. Tabelis on väite jaoks märgitud minimaalne ja maksimaalne hinnang. Lisaks on välja toodud väite kõigi vastuste keskmine tulemus. Tulemustest on näha, et vastajad on kõigi väidete jaoks andnud kas keskmise või keskmisest parema hinnangu. Väide, mis puudutas rakenduse töötamist ilma probleemideta, sai kõige madalama skoori. Kuna rakenduse arendus on alles algusfaasis, siis oli sellist tulemust ka oodata. Madaluselt järgmine skoor oli väitel, mis käis filtreerimiste valikute mõistetavuse kohta. See tähendab, et tulevikus tuleks lisada elemente, mis täpsustaksid filtreerimiste valikute otstarvet. Tasub märkida, et üks vastaja andis filtreerimise funktsionaalsuse jaoks soovitusena järgneva kommentaari:

*„Filtreerimine võiks toimuda automaatselt (aka kui sisestada nt aasta siis võiks see filter juba rakenduda), sest kui faile on seal piisavalt palju siis see lihtsustaks oluliselt õige faili leidmist.“*

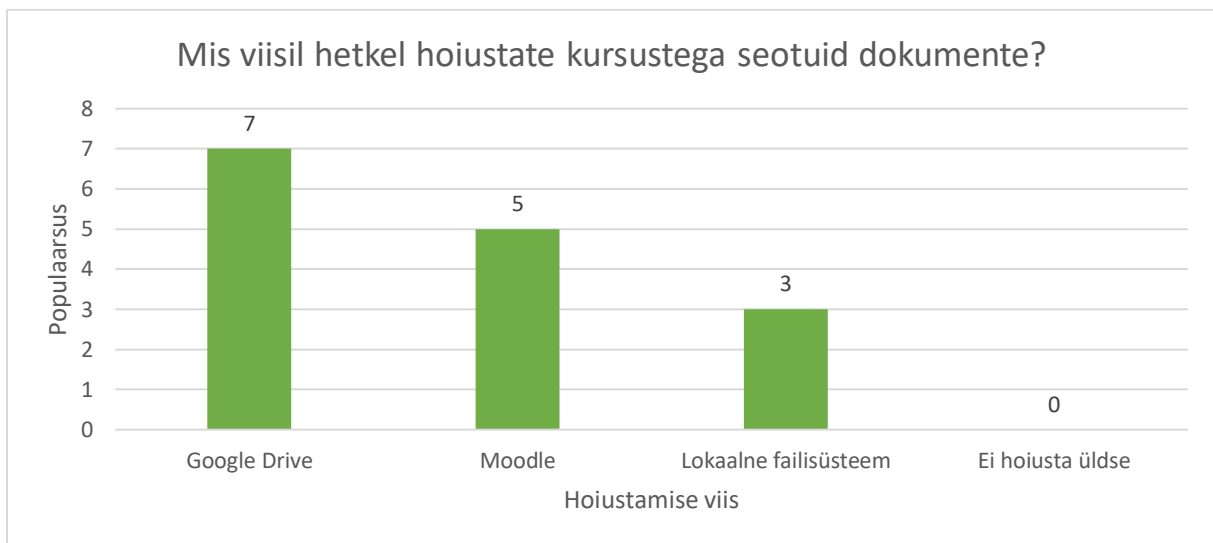
Selline funktsionaalsus oleks kasutajate jaoks mugavam, sest see eemaldab vajaduse filtrit aktiveerida ja deaktiveerida. Taoline lähenemine tuleb aga enne läbi testida. Käesoleva bakalaureusetöö koostamise ajal on keeruline hinnata selle mõju süsteemi kiirusele. Muud väited on hinnatud ühtlaselt ning on saavutanud ülekaalukalt positiivsed tulemused. See näitab, et videolt nähtud funktsionaalsused täidavad oma eesmärgi ja vastavad nõuetele.

Tabel 1. Väidete hinnangute jaotus.

Väide	Min	Max	Keskmine
Uute dokumentide lisamine on arusaadav	4	5	4,8
Dokumentide kustutamine on arusaadav	4	5	4,8
Dokumentide sisu muutmine on arusaadav	4	5	4,8
Dokumentide filtreerimise valikuid on piisavalt	4	5	4,8
Filtreerimiste valikute funktsionaalsused on mõistetavad	3	5	4,3
Navigeerimine on piisavalt intuitiivne	3	5	4,4
Rakenduse stiil on meelepärane	4	5	4,4
Rakenduses olev tekst on sobiva stiili ja suurusega	4	5	4,6
Kasutaja saab tegevuste kohta piisavalt tagasisidet	4	5	4,6
Rakendus töötab probleemideta	3	5	4,2
Vaadete laadimine on piisavalt kiire	4	5	4,7
Rakenduse funktsionaalsused töötavad piisava kiirusega	4	5	4,7

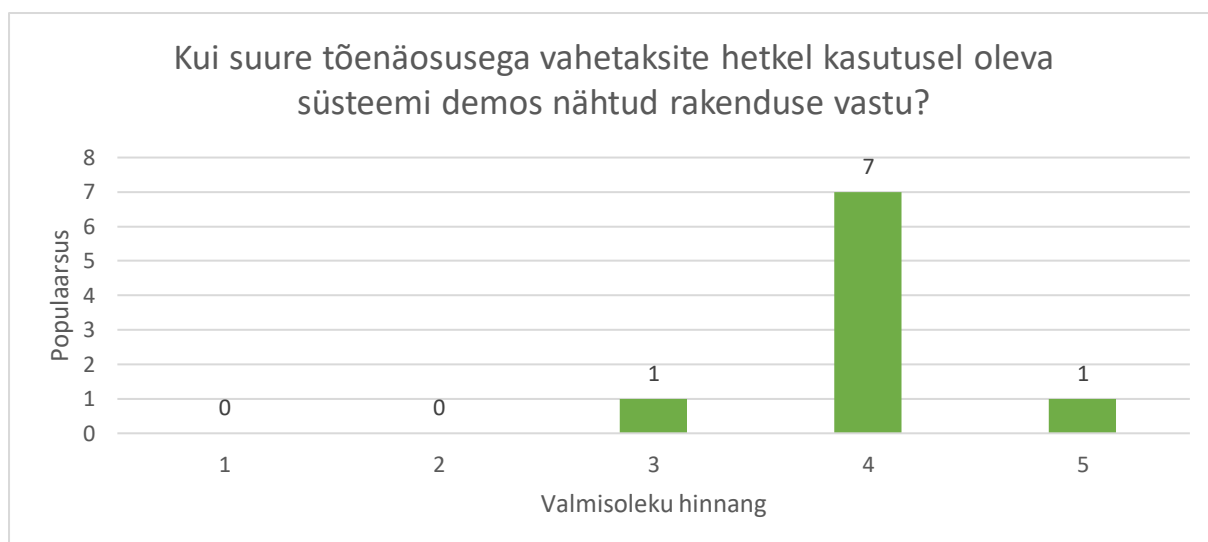
Küsitluse teises pooles uuriti esmalt, millisel viisil vastajad materjale hoiustavad (vt joonis 7). Vastusevariantides oli kokku neli varianti. Valikute seas olid Google Drive, lokaalne

failisüsteem ja Moodle. Välja oli toodud ka võimalus, et vastaja ei hoiustagi faile, kuid seda keegi ei valinud. Vastuste seast on näha, et kõige populaarsemaks osutus Google Drive. Teise ja kolmanda koha saavutasid vastavalt Moodle ja lokaalne failisüsteem. Need kõik variandid pakuvad kasutajatele võimalust faile salvestada, muuta ja alla laadida. Ükski süsteem aga ei võimalda faile filtreerida. Lisaks pole otsest viisi seostada ülesannet ja tema jaoks loodud näidislahendusi ning automaatkontrolle. Järelikult loodud rakendus säilitab väljatoodud süsteemide funktsionaalsused ning lisab lisaväärtusena dokumentide filtreerimise ja failide omavahelise grupeeringu.



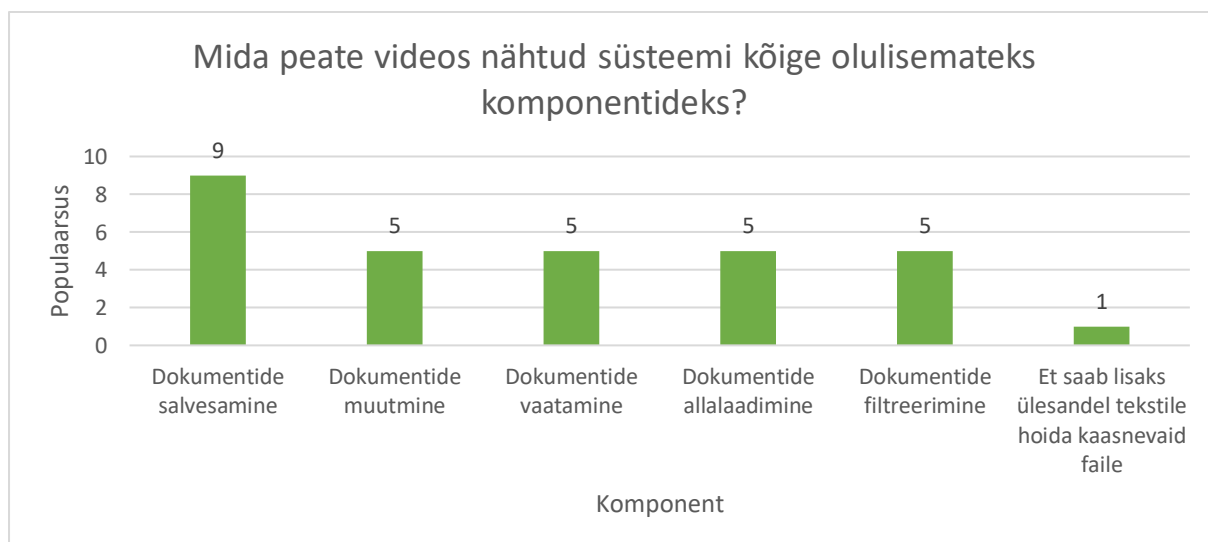
Joonis 7. Vastajate dokumentide hoiustamise viisid.

Järgmisena uuriti küsitluses osalejate käest valmisolekut kasutatav süsteem välja vahetada loodud rakenduse vastu (vt joonis 8). Lisaks valmisolekule hinnangu andmisele paluti vastajatel oma hinnangut ka põhjendada. Vastajad said oma valmisolekut muudatuse tegemiseks hinnata viie palli skaalal, kus madalaim väärtus oli 1 ning kõrgeim hinnang 5. Madalam hinnang tähendab, et vastaja pole uue süsteemi kasutamisest huvitatud. Kõrgem hinnang tähendab aga, et vastaja oleks valmis uut süsteemi aktiivselt kasutama. Vastused on jäänud kas keskmise või keskmisest kõrgema juurde. See näitab, et küsitluses osalejad on rakendusest huvitatud ning on potentsiaalselt valmis oma dokumentide haldamise üle viima loodud rakenduse keskkonda. Vastajate põhjendustest on näha, et peamine põhjus ülemineku tegemiseks on failide filtreerimise ja otsimise funktsionaalsused.



Joonis 8. Vastajate valmisolek süsteemi kasutama hakata.

Vastajate käest uuriti ka, mis nende arvates on rakenduse kõige olulisemad funktsionaalsused. Võimalikud vastused olid dokumentide salvestamine, muutmine, vaatamine, allalaadimine ja filtreerimine. Lisaks oli võimalus vastajal omalt poolt valitud funktsionaalsus välja tuua. Üks vastajatest tõi välja, et oluline komponent süsteemi juures on see, et rakendus võimaldab lisaks ülesande tekstile hoiustada ka ülesandele vastavaid näidislahendusi ja automaatkontrolle. Küsimuse eesmärgiks oli välja selgitada, kas rakenduses olevad funktsionaalsused omavad kasutajate jaoks väärtust. Vastuste hulgast on näha, et kõik funktsionaalsused on vastajate arvates vajalikud (vt joonis 9). See tähendab, et edasiarenduste plaanide puhul tuleks keskenduda kõikide rakenduse aspektidele, mis puudutavad väljatoodud funktsionaalsusi.



Joonis 9. Rakenduse kõige olulisemad komponendid vastajate arvates.

Küsitluse lõpetuseks paluti vastajatel välja tuua, mis neile videos nähtud rakenduse juures meeldis ja ei meeldinud. Lisaks oli võimalus lisada täiendavaid kommentaare, kuidas rakendust

paremaks saaks muuta. Järgnevalt on välja toodud paar vastajate vastust, mis küsitluses osalejatel süsteemi juures meeldis:

*„Rakendus tundus väga kergesti kasutatav ja intuiitiivne. Võrreldes Google Driveiga on see kindlasti palju paremini sobiv erinevate dokumentide hoidmiseks. Väga suureks plussiks tundus olevat see, et rakenduses endas saab kohe dokumente ka muuta (et ei ole tarvidust faili alla laadida, muuta ja siis uuesti üles laadida).“*

*„Meeldis, et oli mitmekülgseid võimalusi alates märksõnadest, filtreerimisest kuni muutmiseni välja. Samuti on väljanägemine nõ puhas ja ei ole infoga üle koormatud.“*

Väljatoodud kommentaarid kinnitavad, et loodud rakenduse funktsionaalsuste olemasolu süsteemis on põhjendatud. Lisaks tuleb välja, et rakenduses kasutatav stiil ja kujundus on sobilik. Kuna kasutajatel polnud võimalik küsitluse läbiviimise hetkel süsteemi ise katsetada, siis oli mittemeeldivate aspektide kirjapanek keeruline. Sellegipoolest tõid mõned vastajad välja huvitavaid tähelepanekuid ja soovitusi. Dokumentide haldamise poole pealt toodi välja, et süsteem võiks võimaldada ka teiste failivormingute lisamist. Pakuti välja ka, et süsteem võiks salvestada failide varasemaid versioone, et neid võimalusel taastada. Nende kommentaaride abil saab rakenduse arendust tulevikus paremini planeerida.

### **3.2 Tuleviku arendusplaanid**

Järgnev peatükk tutvustab rakenduse tuleviku arendusplaane. Arendusplaanid põhinevad käesoleva bakalaureusetöö autori, juhendajate ja tagasisideküsitlusele vastajate ideedel. Peatükis kirjeldatud arendused võetakse ette pärast bakalaureusetöö lõppu.

Esimeseks fookuseks saab olema dokumentide sisu teisendamise täpsuse tõstmine. Hetkel on süsteemi suurimaks miinuseks see, et tagasisüsteem ei suuda eesliidese poolt saadetud dokumendi sisu piisavalt täpselt teisendada .docx failiks nii, et tulemusfailis säiliks eesliideses kuvatud stiil ja vorming. Töö tegemise lõpuks on süsteem võimeline teisendama lihtsamaid elemente nagu näiteks linke, tabeleid ja loetelusid. Seda aga tingimusel, et elemendid on dokumendis esitatud iseseisvalt. Vigu võib esineda, kui tabelis esineb tekst, mis sisaldab linki. Sellisel juhul säilitatakse tabelis küll tekst, aga link ilmub tabelist väljaspool. Taolised ebatäpsused võivad kasutajaid segadusse ajada ja halvimal juhul võivad tehtud muudatused kaotsi minna. Lisaks .docx failide sisu teisendamise täpsuse tõstmisele on plaanis lisada süsteemi veel failitüüpe. Muudeks populaarseteks failivorminguteks on näiteks .xlsx ja .pdf

failid. Enne nende lisamist süsteemi tuleb aga algul kindlaks teha, et süsteem on võimeline nende sisu sobivale kujule teisendama ja kasutajate muudatusi säilitama. Tulevikus on eesmärk süsteemi juurde lisada ka rohkem programmeerimiskeelte lähtekoodifailide tüüpe. Nii on ka teistel kursustel suurem motivatsioon veebirakendust kasutama hakata. Plaan on hakata salvestama dokumentide erinevaid versioone, et kasutajad saaksid neid soovi korral taastada.

Teiseks oluliseks eesmärgiks on juurde lisada dokumentide filtreerimisvõimalusi. Kui rakendus peaks populaarseks osutuma, siis üleslaetavate failide kogus võib kiiresti kasvada. Sellisel juhul ei pruugi võimalik olla olemasolevate võimaluste kaudu piisavalt täpseid otsinguid teha. Selleks, et kasutajate jaoks otsinguaega vähendada, tuleks juurde lisada filtreerimisvalikuid. Näiteks on plaanis lisada dokumendi otsimine läbi dokumendi enda sisu. See tähendab, et kasutajatel on võimalik sisestada paar võtmesõna ning nende kaudu leitakse üles kõik dokumendid, mille sisus leidub sisestatud võtmesõnu. On plaan läbi viia koormusteste, mis võimaldaksid kindlaks teha, kas süsteem peaks vastu, kui filtri lisamisel rakendatakse filter koheselt. Lisaks filtreerimisvalikute lisamisele on plaan teha filtreeritud dokumentide loetelu jaoks erinevad sorteerimisvõimalused. Esialgu on plaan piirduda lihtsamate variantidega. Näiteks võib teostada sorteerimise tähestiku või dokumendile määratud aasta järgi. Sorteerimine oleks suureks abiks kasutajatele, sest selle abil on võimalik vähendada dokumendi otsinguaega.

Kolmandaks plaaniks on süsteemi juurde lisada rolle, millel oleksid erinevad privileegid. Kuna kursus „Programmeerimine” teeb koostööd ka Eesti koolidega<sup>13</sup>, siis on soov lisada süsteemi kasutajatüüp, kellel oleks võimalus dokumente ainult vaadata. Koolidel tekib niiviisi parem arusaam, mida programmeerimise õpetamise juures käsitletakse. Lisaks on võimalik saada inspiratsiooni ülesannete koostamise jaoks. Vaatajarollis olev kasutaja saaks süsteemile ligi, aga dokumente muuta ega kustutada ei saaks. Lisaks funktsionaalsetele piirangutele oleks hädavajalik rollidele juurde lisada privileegid, mis aitaksid kindlaks teha, millistele kursustele kasutajal ligipääs olemas on. Kuna ainetel on tavapärane kaasata tudengitest õppeassistente, siis ei tohiks igal kasutajal kõikidele süsteemis olevatele kursuste dokumentidele ligipääsu olla. Privileegide lisamine kasutajatele tõstaks teiste kursuste motivatsiooni süsteemiga liituda.

---

<sup>13</sup> Vestlusest õppeaine juhtiva õppejõuga.

Eelnevad eesmärgid on ainult mõned tugipunktid, millest lähtutakse, et süsteemi edasi arendada. Plaanis on ka teha muid väiksemaid parandusi nii funktsionaalsuse kui ka stiili ja kujunduse juures, et süsteemi kasutajate jaoks meeldivamaks muuta.

## Kokkuvõte

Bakalaureusetöö eesmärk oli koostada aine „Programmeerimine” jaoks veebirakendus, mis hõlbustaks kursuse õppejõududel kursusega seotuid dokumentide haldamist. Enne arendamist uuriti veebirakenduse olemust ja selle komponente. Tutvustuse tulemusena sai selgeks, et veebirakendus koosneb kolmest osast, milleks on andmetasand, tagasüsteem ja eesliides. Enne kui süsteemi arendamise juurde mindi, selgitati välja, mis tehnoloogiaid arendamiseks kasutama hakatakse. Analüüsi tulemusel valiti andmetasandi, tagasüsteemi ja eesliidese arendamiseks vastavalt PostgreSQL, Spring Boot ja Angular.

Selleks, et arendusprotsess võimalikult tõhusalt toimuks, tehti enne arendamist süsteemi jaoks põhjalik nõueteanalüüs kasutades FURPS metoodikat. Nõueteanalüüsi tulemusena tehti kindlaks rakenduse põhifunktsionaalsused ja muud nõuded, mis puudutasid rakenduse kasutatavust, usaldusväärsust, jõudlust ja toetatavust. Põhifunktsionaalsuste hulka kuulusid dokumentide lisamine, muutmine, kustutamine ja allalaadimise võimalused. Kasutatavuse nõuete abil tehti kindlaks, et rakenduse kasutamine peab jääma intuitiivseks ja kättesaadavaks ka vaegnägijatele. Usaldusväärsuse analüüsist tehti kindlaks, et süsteem ei tohi tõrgete korral kasutuskõlbmatuks muutuda. Jõudluse aspektidest järeldati, et kasutajate päringute ooteaeg peab olema optimaalne. Toetatavuse analüüsi järel sai selgeks, et süsteem peab olema ka arendatav peale käesoleva bakalaureusetöö lõppu.

Pärast nõueteanalüüsi koostamist mindi edasi süsteemi arenduse juurde. Arendati kõiki kolme veebirakenduse süsteemi, milleks olid andmetasand, tagasüsteem ja eesliides. Andmetasandi puhul pandi nõueteanalüüsi alusel paika vajaminevad andmed ja nendevahelised relatsioonid. Tagasüsteem kujunes välja seitsme põhimooduli koostamisel, milleks olid failimoodul, ülesandemoodul, juhendimoodul, kasutajamoodul, autentimisemoodul, märksõnamoodul ja kursusemoodul. Lisaks toodi välja erinevad ülesanded, mille eest tagasüsteem hoolitseb, et suhtlus eesliidese ja andmetasandi vahel toimiks. Eesliidese töö jaotati laiali kolme põhimooduli abil. Need kolm moodulit olid peamoodul, avalik moodul ja kaitstud moodul. Lisaks eesliidese töö jaotusele anti ülevaade rakenduse põhivaadetest. Põhivaated sisaldasid registreerimise, sisselogimise, dokumentide loetelu ja kasutajate haldamise vaateid. Lisaks näidati, et rakendus on kättesaadav vaegnägijate jaoks.



Rakenduse arendustöö lõpuks koostati tagasisideküsitlus. Läbiviidud küsitluse abil tehti kindlaks, et loodud rakendus vastab nõuete analüüsis paika pandud kriteeriumitele. Selgitati välja, et küsitluses osalenud inimestel on huvi loodud veebirakendust kasutama hakata. Tehti kindlaks, et arendatud süsteem säilitab vastanute jaoks olulised funktsionaalsused ja lisab juurde ka lisaväärtusi. Viimaseks koguti küsitluse abil mõtteid edasiarenduse plaanideks. Lisaks tagasiside analüüsile pandi paika ka tulevikuplaanid. Edasised arendusplaanid sisaldasid rakenduse kvaliteeti tõstvaid aspekte, mis võetakse arendusse peale bakalaureusetöö lõppu.

## Viidatud kirjandus

- [1] Mohamad Gharib. Web Applications – Basic concepts. 2022. [https://courses.cs.ut.ee/LTAT.05.004/2022\\_fall/uploads/Main/Web%20Applications%20%E2%80%93%20Basic%20concepts.pdf](https://courses.cs.ut.ee/LTAT.05.004/2022_fall/uploads/Main/Web%20Applications%20%E2%80%93%20Basic%20concepts.pdf) (10.12.2022)
- [2] Getting Started – About Version Control. <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control> (10.12.2022)
- [3] Developer survey. 2021. <https://insights.stackoverflow.com/survey/2021#overview> (10.12.2022)
- [4] DB-Engines ranking. <https://db-engines.com/en/ranking> (10.12.2022)
- [5] Himanshu Kathuria. The Most Popular Databases for 2022. 2022. <https://learnsql.com/blog/most-popular-databases-2022> (10.12.2022)
- [6] Ali Spittel. What is a Web Framework, and Why Should I use one? <https://welearncode.com/what-are-frontend-frameworks> (03.03.2023)
- [7] Natallia Sakovich. The Most Popular Frontend Frameworks 2023. 2023. <https://www.sam-solutions.com/blog/best-frontend-framework> (03.03.2023)
- [8] Ahead-of-time (AOT) compilation. 2022. <https://angular.io/guide/aot-compiler> (03.03.2023)
- [9] Daniel Fleury. Who Uses Angular in 2023? 12 Global Websites Built With Angular. 2023. <https://www.trio.dev/blog/companies-use-angular> (03.03.2023)
- [10] Introduction to Angular concepts. 2022. <https://angular.io/guide/architecture> (03.03.2023)
- [11] Sakshi Pandey. The importance Of Code Reusability In Software Development. 2022. <https://www.browserstack.com/guide/importance-of-code-reusability> (03.03.2023)
- [12] The most popular JavaScript frameworks and libraries. 2023. <https://www.ionos.com/digitalguide/websites/web-development/popular-javascript-frameworks-and-libraries> (03.03.2023)
- [13] Jon Fincher and Kim Desmond. The 11 Best Backend Frameworks – 2023. 2023. <https://codingnomads.co/blog/best-backend-frameworks> (06.03.2023)
- [14] Spring Boot Architecture. <https://www.javatpoint.com/spring-boot-architecture> (06.03.2023)
- [15] Spring Security. <https://docs.spring.io/spring-security/reference/index.html> (06.03.2023)
- [16] Thorben Janssen. What is Spring Data JPA? And why should you use it? <https://thorben-janssen.com/what-is-spring-data-jpa-and-why-should-you-use-it> (06.03.2023)
- [17] Spring Boot Caching. <https://www.javatpoint.com/spring-boot-caching> (06.03.2023)

- [18] The Most Popular Programming Languages – 1965/2022 – New Update. <https://statisticsanddata.org/data/the-most-popular-programming-languages-1965-2022-new-update> (06.03.2023)
- [19] Robert C. Martin. Clean Code: A Handbook of Agile Software Craftsmanship. 1<sup>st</sup> Edition. Pearson. 2008.
- [20] Functional and Nonfunctional Requirements: Specification and Types. 2021. <https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types> (08.03.2023)
- [21] Alex Papworth. Non Functional Requirements. <https://businessanalystmentor.com/non-functional-requirements> (08.03.2023)
- [22] Blindness and vision impairment. 2022. <https://www.who.int/en/news-room/fact-sheets/detail/blindness-and-visual-impairment> (08.03.2023)
- [23] Color Blindness In Clinton. <https://www.clintoneye.com/color-blindness.html> (08.03.2023)
- [24] Top Browsers Market Share. 2023. <https://www.similarweb.com/browsers> (08.03.2023)
- [25] What are Web Standards. 2015. <https://www.elcom.com.au/resources/blog/web-standards> (08.03.2023)
- [26] Michael Wiegand. Site Speed is (Still) Impacting Your Conversion Rate. 2022. <https://www.portent.com/blog/analytics/research-site-speed-hurting-everyones-revenue.htm> (08.03.2023)
- [27] Response Time. <https://sematext.com/glossary/response-time> (08.03.2023)
- [28] Spring Security. <https://docs.spring.io/spring-security/reference/index.html> (28.04.2023)
- [29] Introduction to modules. 2022. <https://angular.io/guide/architecture-modules> (30.04.2023)

## **Lisad**

### **I. Valminud veebirakenduse lähtekood**

Veebirakenduse lähtekood ja lokaalselt ülesseadmise juhend:

<https://github.com/KaurTaal/hades>

## II. Tagasiside saamiseks saadetud küsitlus

Käesoleva küsitluse eesmärgiks on koguda tagasisidet programmeerimisülesannete infosüsteemile. Selleks, et küsimustele vastata, tuleb eelnevalt läbi vaadata demovideo, mis tutvustab rakenduses olevaid funktsionaalsusi.

Kuna rakenduse ülespanek veebi on kujunenud pikemaks protsessiks, kui seda algul arvati, siis on tagasisideküsitluse jaoks kasutatud videot. Demovideo asub järgneval lingil: <https://drive.google.com/drive/folders/1AwgOUZws0DTr8FVKd2H5-49hraI2K3N0>.

Tagasisideküsitluse viib läbi Tartu Ülikooli tudeng Kaur Taal oma informaatika eriala bakalaureusetöö raames. Video läbivaatamine ja ankeedi täitmine võtab aega umbes 15 minutit. Käesolev küsitlus tagab vastajatele täieliku konfidentsiaalsuse ja küsitluse käigus kogutud andmeid kasutatakse ainult kokkuvõtlikult.

Küsimuste korral võib pöörduda autori poole aadressil [küsitluse läbiviija kontakt](#).

Veebirakenduse elementide hindamine, mida hinnati viie palli skaalal:

1. Uute dokumentide lisamine on arusaadav.
2. Dokumentide kustutamine on arusaadav.
3. Dokumentide sisu muutmine on arusaadav.
4. Dokumentide filtreerimise valikuid on piisavalt.
5. Filtreerimise valikute funktsionaalsused on mõistetavad.
6. Navigeerimine on piisavalt intuitiivne.
7. Rakenduse stiil on meelepärane.
8. Rakenduses olev tekst on sobiva stiili ja suurusega.
9. Kasutaja saab tegevuste kohta piisavalt tagasisidet.
10. Rakendus töötab probleemideta.
11. Vaadete laadimine on piisavalt kiire.
12. Rakenduse funktsionaalsused töötavad piisava kiirusega.

Soovi korral lisa väidete kohta täpsustavaid kommentaare.

Küsimused veebilehe kasutamise kohta:

1. Mis viisil hoiustate kursustega seotuid dokumente? (mitme vastusega küsimus)

- a. Google Drive
  - b. Lokaalne failisüsteem
  - c. Ei hoiusta üldse
  - d. Moodle
  - e. Other...
2. Kui suure tõenäosusega vahetaksite hetkel kasutusel oleva süsteemi demos nähtud rakenduse vastu? (viie palli skaala küsimus)
3. Palun põhjendage eelnevat vastust.
4. Mida peate videos nähtud süsteemi kõige olulisemateks komponentideks? (mitme vastusega küsimus)
- a. Dokumentide salvestamine
  - b. Dokumentide muutmine
  - c. Dokumentide vaatamine
  - d. Dokumentide allalaadimine
  - e. Dokumentide filtreerimine
  - f. Other...
5. Mis Teile rakenduse juures meeldis?
6. Mis Teile rakenduse juures ei meeldinud?

Soovi korral lisage kommentaare, mida võiks rakenduse juures paremaks teha.

### **III. Tagasisideküsitluses kasutatud demovideo**

Demovideo Google Drive'i keskkonnas:

<https://drive.google.com/drive/folders/1AwgOUZws0DTr8FVKd2H5-49hraI2K3N0>

# Litsents

## Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, **Kaur Taal** (sünnikuupäev: 07.08.2001),

1. Annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose **Veebirakenduse loomine kursuse „Programmeerimine“ programmeerimisülesannete haldamiseks**, mille juhendajad on **Tauno Palts** ja **Reimo Palm**, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Kaur Taal

Tartus, **09.05.2023**