

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Robert Pärnpuu 206053IABB

**TALTECH ÜLIÕPILASESINDUSE
RAHASTUSPLATVORMI ARUANNETE
GENEREERIMISE FUNKTSIONAALSUSE
ARENDUS**

Bakalaureusetöö

Juhendaja: Tarvo Treier
MSc

Tallinn 2023

Autorideklaratsioon

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

Autor: Robert Pärnpuu

17.05.2023

Annotatsioon

Dokumentide loomine on tavaline osa organisatsioonide tööst, kuid suure koguse andmete korral võib dokumentide käsitsi loomine olla äärmiselt ajakulukas. Andmete hoiustamine digitaalselt pakub alternatiivi, dokumentide automaatset genereerimist.

Sellist lahendust soovis ka TalTechi üliõpilasesindus, mis hoiustab andmeid tudengielu rahastamise kohta varem loodud rahastusplatvormil ning peab aruandvatel eesmärkidel esitama andmeid PDF failitüübis seletuskirjades. Käesoleva töö eesmärgiks on kirjeldatud lahenduse loomine.

Eesmärgi lahendamiseks analüüsib autor probleemi ning erinevaid dokumentide automaatset genereerimist võimaldavaid tööriistu ja nende kasutusega seotud omadusi. Korrekse lahenduse loomiseks kogutakse üliõpilasesinduselt ja erinevatelt tarkvaraarenduse põhiprintsiipidelt nõudeid.

Kogutud teadmiste põhjal saab loodud nõuetele vastav lahendus ning kirjeldatud selle loomisega seoses tehtud valikuid, muudatusi ja lahenduse üldist tööprotsessi. Lahendus saab valideeritud kliendi poolt ning vajalikud edasiarendused saavad kirja pandud.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 30 leheküljel, 7 peatükki, 8 joonist, 3 tabelit.

Abstract

Development of Report Generation Functionality for TalTech Student Union's Funding Platform

Nearly every organization and business will eventually have the need for creating documents for a variety of different purposes. When having to deal with a large amount of data, creating documents by hand will turn out to be very time consuming. Digital storage of data offers a solution to this problem, automating the generation of documents.

TalTech Student Union ran into the aforementioned problem with its funding platform, which is used to fund student life in the university. The Student Union must create explanatory letters for all its funding contests for accountability reasons and they wish to automate that process. The goal of this thesis is to develop functionality that enables that.

In order to create the solution, the author analyzes the problem and different tools that enable automatic PDF document generation. In addition, they collect different requirements from the Student Union and widely accepted software development guidelines.

With that information, a solution is created that fits the set requirements and enables the creation of other documents as well. Related decisions, made changes and the inner workings of it are documented. The result is validated by the client and needed developments are documented as well.

The thesis is in Estonian and contains 30 pages of text, 7 chapters, 8 figures, 3 tables.

Lühendite ja mõistete sõnastik

AGPL	<i>GNU Affero General Public License</i>
API	<i>Application programming interface</i> , rakendusliides
BSD	<i>Berkeley Source Distribution</i>
<i>Copyleft</i>	Viis tarkvara vaba kasutuse võimaldamiseks, mis kohustab nende muudetud versioonide jagamisel hoida samu vabadusi.
CSS	<i>Cascading Style Sheets</i> , kaskaadlaadistik, keel märgistuskeelse dokumendi esitluse kirjeldamiseks.
DOCX	Tekstitötluse failitüüp peamiselt Microsoft Wordile.
<i>Gang of four</i>	Raamatu „Design Patterns: Elements of Reusable Object-Oriented Software“ autoritele viitamiseks kasutatud nimi.
<i>Git fork</i>	Koodibaasi kloonimise viis kasutades Git versioonihaldustarkvara.
HTML	<i>HyperText Markup Language</i> , hüperteksti märgistuskeel, levinuim veebidokumentide kodeerimissüsteem.
HTTP	<i>Hypertext Transfer Protocol</i> , hüperteksti edastuse protokoll, protokoll veebi kaudu failide edastamiseks.
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IT	Infotehnoloogia
JAR	<i>Java archive</i> , failitüüp Java rakendusega seotud failide levitamiseks.
JSON	<i>JavaScript Object Notation</i> , JavaScripti objektide notatsioon, inim- ja masinloetav formaat andmete vahendamiseks.
JWT	<i>JSON Web Token</i> , standard turvalise andmevahenduse võimaldamiseks.
LGPL	<i>GNU Lesser General Public License</i>
MIT	<i>Massachusetts Institute of Technology</i>
MPL	<i>Mozilla Public License</i>
ODT	OpenDocument tekstitötluse failitüüp.
PDF	<i>Portable Document Format</i> , porditav dokumendivorming, välimust ja sisu hoidev failiformaat.
<i>Permissive</i>	Tarkvara litsentside tüüp, mis piirab nende kasutust, muutmist ja jagamist minimaalselt.

SaaS	<i>Software as a Service</i> , tarkvara teenusena, tarkvara tarnimise meetod, millega pakutakse tarkvara pilvepõhist kasutust teenusena.
<i>Single Page Application</i>	Veebirakenduse tüüp, mis laeb korraga terve veebidokumendi ning uuendub uute andmetega.
SRS	<i>Software requirements specification</i> , tarkvara nõuete spetsifikatsioon, arendatava tarkvara eelduste kogum.
<i>Token</i>	Volitustõend, objekt, mis tõendab toimingute sooritamise õigust.
URI	<i>Unified Resource Identifier</i> , ühtne ressursi-identifikaator, ressursi identifitseeriv märgistring
XHTML	<i>Extensible HyperText Markup Language</i> , laiendatav hüperteksti märgistuskeel, XML süntaksi reeglitele vastav HTML fail.
XML	<i>Extensible Markup Language</i> , laiendatav märgistuskeel, inim- ja masinloetav märgistuskeel, millega hoiustatakse ja edastatakse andmeid.

Sisukord

1 Sissejuhatus	11
2 Probleemi püstitus	13
2.1 Kasutusel olev protsess.....	13
2.2 Aruannete elemendid	14
2.3 Seletuskirja struktuur	16
3 Dokumentide genereerimise rakenduste tüübid	18
3.1 Generaatorite tüübid	18
3.1.1 Mallipõhised generaatorid	18
3.1.2 Märgistuskeelte konverterid	20
3.1.3 Dokumentide ehitamise teegid	20
3.2 Generaatori tüübi valimine	21
3.3 Kasutusõigus.....	22
3.3.1 MIT License	23
3.3.2 BSD License	23
3.3.3 Apache License 2.0	23
3.3.4 Mozilla Public License 2.0	23
3.3.5 GNU Lesser General Public License.....	24
3.3.6 GNU Affero General Public License	24
3.4 Generaatorite võrdlus	24
4 Nõuete kogumine.....	26
4.1 Nõuete küsimused.....	26
4.2 Nõuded kliendilt	27
4.2.1 Kasutajalugu	28
4.3 Tehnilised nõuded.....	29
4.4 Olemasoleva rakenduse nõuded	30
5 Arendatud lahendus	31
5.1 Generaatori valik	31
5.1.1 Mallipõhised generaatorid	31
5.2 Lahenduse struktuur	32

5.3 Muudatused kasutajaliideses	32
5.4 Muudatused rakendusliideses	33
5.5 Dokumendi ehitaja kasutamine	35
6 Tulemused ja analüüs	37
6.1 Tulemuste valideerimine	37
6.1.1 Nõuetele vastavus	37
6.2 Edasiarendus	38
7 Kokkuvõte	40
Kasutatud kirjandus	41
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	44
Lisa 2 – Käsitsi loodud seletuskirja näidis	45
Lisa 3 – Genereeritud seletuskirja näidis.....	49

Jooniste loetelu

Joonis 1. Konkurssi tulemuste vaade rahastusplatvormil.....	14
Joonis 2. Spetsiifilise taotluse tulemust kokku võttev sektor.	17
Joonis 3. Näide dokumentide genereeriva teenuse arve mallist [4].	19
Joonis 4. Näide iText 7 Core abil dokumendi ehitamisest	21
Joonis 5. Konkurssi tulemuste vaade.....	33
Joonis 6. Rakendusliideses seletuskirja genereerimise protsessi diagramm.	35
Joonis 7. ReportBuilder loomise ja dokumendi omaduste konfigureerimise näide.	35
Joonis 8. ReportBuilder kasutamise näide esilehe loomiseks.	36

Tabelite loetelu

Tabel 1. Dokumendi standardsed elemendid.....	16
Tabel 2. Dokumentide genereerimise teegid ja teenused ning nende omadused [5]–[11], [14], [15], [17], [19], [21], [23], [26], [27], [28].	25
Tabel 3. Dokumendi genereerimise päring.	33

1 Sissejuhatus

Pea kõigil tegelevatel ettevõtetel ja organisatsioonidel tuleb varem või hiljem ette vajadus koguda andmeid dokumentidesse, olgu need füüsilise paberi või tänapäeval levinud digitaalsete failide kujul. Neid võib minna vaja minna näiteks finantstehingute dokumenteerimiseks, informatsiooni jagamiseks ja hoiustamiseks, lepingute sõlmimiseks ning paljudeks muudeks eesmärkideks. Vahel vajavad dokumendid otsest inimese poolset sisendit, kuid tihti koosnevad dokumendid vaid kindlatest andmetest, mille korral muutub käsitsi loomine ebavajalikuks ning suurte koguste korral äärmiselt ajakulukaks. Andmete digitaalne hoiustamine andmebaasides pakub alternatiivi, automatiseeritud loomist, kuid selle saavutamiseks on vaja luua lahendusi, mis seda võimaldaksid.

Paljudel äritarkvaradel leiduvad funktsionaalsused vajalike dokumentide automaatseks genereerimiseks, mis vastavad klientide vajadustele, kuid majasiseselt loodud tarkvarade puhul peab vastavaid lahendusi sisse ostma või ise looma. Abiks olevate teenuste ja teekide suure koguse tõttu võib lahenduse loomiseks parima tööriista leidmine võtta lahenduse loojate poolt arvestavalt uurimistööd.

Aruandvate dokumentide manuaalse loomise probleemide ette sattus Tallinna Tehnikaülikooli üliõpilasesindus (edaspidi üliõpilasesindus), mis kasutab Krõõt Grete Mändi [1] ja Raimond Lume [2] bakalaureusetööde raames loodud rahastusplatvormi selleks, et võimaldada ülikooli organisatsioonidel ja eraisikutel taotleda rahastust erinevate projektide läbiviimiseks ja organisatsioonide kulude katmiseks.

Lõputöö eesmärk on luua rahastusplatvormile üliõpilasesinduse poolt esitatud nõuetele ja kvaliteetse tarkvaraarenduse põhiprintsiipidele vastava aruannete genereerimise funktsionaalsuse.

Eesmärgi saavutamiseks analüüsib autor praegu kasutusel olevaid protsesse ning toob välja lahendatava probleemi omadusi. Probleemi lahendamiseks uuritakse erinevaid PDF dokumentide genereerimist võimaldavaid teenuseid ja teeke, võttes arvesse nende kasutusega seonduvaid omadusi. Täpsema lahenduse plaani paika panekuks kogub autor nõudeid üliõpilasesinduselt, platvormi tehnilistelt omadustelt ning tarkvara arenduse

põhiprintsiipidelt. Seejärel kirjeldatakse tehtud valikuid, muudatusi rahastusplatvormile ning lahenduse tööprotsessi. Töö lõpeb tulemuste analüüsiga ning arutletakse edasise samme lahenduse arenguks.

2 Probleemi püstitus

Käesolevas peatükis analüüsib autor praeguseid üliõpilasesinduse konkursside aruannete loomise protsesse ning selleks kasutatud dokumendi struktuuri, mille eesmärgiks on panna paika lahendatava probleemi omadused ning eeldused lahenduse jaoks.

2.1 Kasutusel olev protsess

Hetkel kasutusel oleva protsessi käigus loob üliõpilasesinduse rahastuskomisjoni esimees aruandvatel ning informatsiooni hoiustavatel eesmärkidel konkursside tulemuste seletuskirjad. Kuna konkursside läbiviimiseks on kasutusel rahastusplatvorm, leidub kõik konkursside, taotluste ja hinnangutega seotud informatsioon seal. Seletuskirjaks vajalik informatsioon hindamiste ja eelarvete kohta on kokku võetud konkursi tulemuste vaates, mille näide on kujutatud joonisel 1. Andmed sisestatakse käsitsi DOCX failitüübis malli, mille põhjal salvestatakse valminud fail PDF failitüüpi ja arhiveeritakse peale digiallkirjastamist üliõpilasesinduse failihaldussüsteemi. Seletuskirja näide on leitav Lisas 2.

PDF failitüüp on valminud dokumendi eelduseks, kuna see kodeerib kõik andmed originaalses faili formaadis, kindlustades andmete ja välimuse ühtsuse igas seadmes. Nende omaduste tõttu on see enimkasutatud failitüüp dokumentide hoiustamiseks ja jagamiseks [3].

Suured projektid mai 2023

Tulemused

€ Konkursi eelarve
10000€

Σ Taotletud toetused kokku
3450€

Σ Välja jagatud toetused kokku
2650€

Esitatud taotlused

Teaduse Konverents
TuxÜK
Keskmine hinne: 81

Otsus

Seletus
Teaduse Konverents on hästi läbi mõeldud üritus koos kogenenud korraldusmeeskonnaga. Ürituse eesmärgid ülikoolis teadusega tegelevate institutsioonide turundamiseks tudengitele on õilis eesmärk ning väga toetatud. Eelarves probleeme pole ning hindamiskomisjon toetab projekti täies mahus.

Summade eraldamine

Toetatud täies mahus

#	Kululiik	Toetuseks taotletav summa	Eraldatud summa
1	Saali rent	200 €	200 €
2	Esinejate kingitused	50 €	50 €
3	Toitlustus	500 €	500 €
4	Turundus	100 €	100 €
Kokku:		850 €	850 €

▼ Näita hindamise tulemusi

Tudengi Olümpia 2023
Juuliusse Sportlased
Keskmine hinne: 78

TalTech Festival
TuxÜK
Keskmine hinne: 53

Joonis 1. Konkurssi tulemuste vaade rahastusplatvormil

Antud protsessi nõrk külge on selle ajakulu. Suure koguse taotluste tõttu leidub palju andmeid, mis vajavad ümberkirjutamist seletuskirja ning selle käsitsi tegemine on äärmiselt ajakulukas. Selle probleemi tõttu tõstatati soov seda protsessi automatiseerida.

2.2 Aruannete elemendid

Seletuskiri, nagu ligi kõik korduvalt loodavad dokumendid, koosneb nii staatilistest kui ka dünaamilistest elementidest, seega nende lisamise võimekus on eeldus töötavale lahendusele. Staatilised elemendid on need, mis püsivad samasugused iga seletuskirja korral, näited nendest on seletuskirja peatükkide ja loendite pealkirjad ning dokumendi disainelemendid. Dokumendi mall koosneb peamiselt vaid staatilistest elementidest, erandina on vaid kohatäidised, mis asendatakse lõpliku dokumendi loomisel sellele spetsiifiliste andmetega.

Dünaamilised elemendid on sellised, mis muutuvad iga erineva dokumendi korral, need on dokumendis spetsiifilised andmed, näiteks seletuskirjas komisjoni koosseisu liikmed,

kuupäevad, projektide andmed jne. Dokumentide automatiseerimiseks on vajalik võimekus defineerida dünaamilisi elemente ning asendada neid lõplike andmetega.

Peale staatiliste ja dünaamiliste elementide võivad dokumendid erineda struktuuri ja välimuse poolt oluliselt, kuid koosnevad kokkuvõttes paljudest erinevatest elementidest, mis täidavad standardseid eesmärgi.

Tabel 1 loetleb valikut peamistest standardiseeritud elementidest, millest luuakse erinevat tüüpi dokumente. Igal elemendil leidub omadusi, mis võivad erineda mallide vahel ning mille omadusi peaks olema võimalik muuta soovitud malli loomiseks.

Element	Kirjeldus	Varieeruvus
Disaini element	Igal lehel kuvatud element ühtlustatud välimuse tagamiseks, võib olla tekst või pilt, tihtipeale organisatsiooni logo.	Suurus, asukoht, sisu.
Taust	Teiste elementide taga leiduv värv või disain.	Läbipaistmatus, sisu.
Pealkiri	Üldiselt tavalisest tekstist suurem, tihti paks tekst, mitu standardse välimusega taset. Eesmärk jaotada tekst temade kaupa.	Šrift, suurus, stiil, joondus, värv, reavahed, sisu.
Tekst	Sisu teksti vormis	Šrift, suurus, stiil, joondus, värv, reavahed, sisu.
Tühi ala	Elementide eraldamiseks kasutatud sisuta rida või suurem ala.	Suurus.
Pilt	Visuaalseid andmeid esitlev element.	Asukoht, suurus, sisu.
Täpploend	Vertikaalne ilma järjeta nimekiri, iga elemendi ees sama tärn.	Teksti omadused, tase, täрни välimus, sisu.
Numberloend	Järjestatud nimekiri, elementide ees järjenumbr.	Teksti omadused, tase, nummerdatuse stiil, sisu.
Tabel	Lahtritest koosnev andmeid hoiustav kahetasandiline struktuur.	Ridade ja veergude kogus, lahtrite ühendatus, välimus, päiserea / -veeru olemasolu, summa rida, sisu.
Lehekülje number	Lehekülgede järge hoidev element.	Šrift, suurus, stiil, asukoht.

Eraldusjoon	Visuaalne joon erineva sisu eraldamiseks.	Stiil, suurus.
Päis	Dokumendi lehe tipus olev ala, mis hoiustab erinevaid elemente.	Suurus, sisu.
Jalus	Dokumendi lehe põhjas olev ala, mis hoiustab erinevaid elemente	Suurus, sisu.
Diagramm	Andmete visualiseerimiseks loodud diagramm.	Diagrammitüüp, välimus, andmete märgistus, sisu.

Tabel 1. Dokumendi standardsed elemendid.

Peale elementide leidub ka dokumendi üldiseid omadusi, mida peaks saama kasutaja sättida enne dokumendile sisu lisamist:

- Dokumendi formaat
- Vaba ääre suurus
- Faili metaandmed

Täielikult kasutuskõlblik dokumentide generaator peab pakkuma kasutajale võimekuse sättida neid omadusi, luua väljatoodud elemente ning võimaldada kasutajal kohendada neid vastavalt oma vajadustele.

Kasutajakogemuse edendamiseks tuleb kasuks arvestamine elementide alamliikidega. Alamliikideks on peamiste elementide kasutus spetsiifilistes olukordades, spetsiifiliste omadustega. Näiteks saab tuua jooniste ja tabelite seletitlid, mis on üldiselt vaid teksti element, kuid spetsiifiliste omadustega. Nende loomise ja omaduste täpsustamise võimaldamine lihtsustab oluliselt kasutajakogemust.

2.3 Seletuskirja struktuur

Seletuskirja soovitud struktuur on üsna tavaline aruandliku dokumendi jaoks, peale sektsioonide kordamise vajaduse. Rahastuskonkursside seletuskirjad koosnevad nii kokkuvõtivatest tabelitest ja tekstidest kui ka taotluse spetsiifilistest sektsioonidest. Joonis 2 on näide rahastuskomisjoni selgituskirja taotluse kokkuvõttest, mida on vaja korrata iga taotluse kohta.

1. Teaduse Konverents

Teaduse Konverents on hästi läbi mõeldud üritus koos kogenenud korraldusmeeskonnaga. Ürituse eesmärgid ülikoolis teadusega tegelevate institutsioonide turundamiseks tudengitele on õilis eesmärk ning väga toetatud. Eelarves probleeme pole ning hindamiskomisjon toetab projekti täies mahus.

Kululiigid	Taotlev summa	Eraldatud summa
Saali rent	200 €	200 €
Esinejate kingitused	50 €	50 €
Toitlustus	500 €	500 €
Turundus	100 €	100 €
Kokku	850 €	850 €

Komisjon rahuldab taotluse täielikus mahus 850 eurot.

Taotletav: 850 eurot

Eraldatud: 850 eurot

Joonis 2. Spetsiifilise taotluse tulemust kokku võttev sektor.

3 Dokumentide genereerimise rakenduste tüübid

Käesolevas peatükis analüüsib autor erinevat tüüpi teenuseid ja teeke, mis võimaldavad dokumentide automaatset genereerimist, eesmärgiga kaardistada nende omadusi ning valida nende põhjal sobiv tööriist rahastusplatvormi arenduseks.

3.1 Generaatorite tüübid

Dokumentide automaatseks genereerimiseks leidub suur kogus erinevaid teenuseid ja teeke, mis omavad erinevaid funktsionaalsuseid ning varieeruvad oluliselt kasutusviiside vahel.

3.1.1 Mallipõhised generaatorid

Lihtsaim viis dokumentide automaatset genereerimist üles seada on kasutades mallipõhiseid generaatoreid. Nende korral loob kasutaja dokumendi malli, täpsustades ära struktuuri, disaini, staatilised elemendid ning märgistades dünaamiliste andmete asukohta kasutades kohatäiteid. Kohatäited märgistatakse üldiselt ära süsteemi poolt defineeritud spetsiifiliste märkidega. Joonisel 3 on näha osa dokumentide genereerimise teenuse Docmosis arve mallist, kus kohatäited on märgistatud topelt „väiksem kui“ ja „suurem kui“ märkide vahele.



Invoice

<<issuerName>>
 <<issuerAddress1>>
 <<issuerAddress2>>
 Tax Id: <<issuerTaxNumber>>

Previous Periods

<<prev1Date>> <<prev1Amt>>	<<prev2Date>> <<prev2Amt>>
-------------------------------	-------------------------------

To: <<name>>
 <<company>>
 <<addr1>>
 <<addr2>>

Account Summary

Opening Balance	\$<<openingBal>>
<<rr_items>> <<item>>	\$<<amt>>
<hr/>	
This Invoice	\$<<totalAmt>>
Tax included in this amount	\$<<tax>>

All charges shown include TAX unless otherwise stated.

Tax Invoice Issued:	<<invoiceDate>>
Invoice Number:	<<invoiceNo>>
<hr/>	
Customer Number:	<<customerNo>>
<hr/>	
Overdue – Pay Now:	\$<<openingBal>>
<hr/>	
Total Owing:	\$<<totalAmt>>
Due Date:	<<dueDate>>

NOTE: THIS STATEMENT IS FOR YOUR RECORDS ONLY!

Your nominated credit card or bank account will be debited on or around <<debitDate>>

Joonis 3. Näide dokumentide genereeriva teenuse arve mallist [4].

Valmisoleva malli korral edastatakse generaatorile mall ning dünaamiliste elementide nimedele vastavad andmed, mille tulemusteks genereeritakse valmis dokument. Andmete edastamine käib pilvepõhistel teenustel üldiselt läbi API päringute, seoste teiste kasutusel olevate ettevõtte tarkvaradega või läbi otsese seose andmebaasiga. Teekide vormis mallipõhiste generaatorite korral edastatakse andmed läbi koodis muutujate defineerimise.

Mallide kasutus võimaldab kasutajal kiiret ja lihtsat dokumendi struktuuri ja välimuse muutmist. Malli loomise viis oleneb generaatori vajadustest, vahendid nagu Docmosis ja XDocReport kasutavad malliks DOCX või ODT failitüübis malle, mis võimaldavad tihti olemasolevate mallide ümbersättimist ja taaskasutust [5], [6].

Malli disaineritega generaatorid kaasavad sisseehitatud dokumendi disainimise liideseid, mis võimaldavad ühes rakenduses algusest lõpuni dokumendi loomist. Disainerid lisavad mugavust ning lisavõimekuse puhul võivad olla paindlikumad, kuid eraldi rakenduse õppimine ja puudulikud funktsionaalsused võivad tekitada ebavajalikku vaeva. Seda lähenemist kasutavad näiteks iText DITO ja JasperReports [7], [8].

Selliseid lahendusi leidub tihti läbi SaaS ärimudelite, mille näol müüakse pilvepõhise tarkvara kasutust teenusena, kus genereerimise üles seadmine toimub läbi veebirakenduste. Sellised teenused pakuvad mugavat kasutajakogemust ning võivad vajada IT-alaseid teadmisi vaid andmete edastamise viisi ülesseadmisel. Siiski selliste teenuste kasutamisel võib probleemiks tulla finantsiline aspekt, mis raskendaks kasutust väikestele organisatsioonidele ning hobiprojektide loojatele.

Mallipõhiseid generaatoreid leidub ka otse rakenduse koodi tasandil, kasutades teeke andmete sisestamiseks ning lõpliku dokumendi genereerimiseks. Andmete edastamiseks generaatorisse kasutatakse üldiselt paisktabeleid. Sellised lahendusi pakuvad näiteks JasperReports [8] ja XDocReport [6].

3.1.2 Märjistuskeelte konverterid

Märjistuskeelte (ingl. k *markup language*) konverterid näevad ette dokumentide loomise kasutades HTML, XHTML või XML faile ning kujunduse mõjutamist CSS abil. Nende kasutusel saab ära kasutada arendajate seas levinud oskuseid ning märjistuskeelte paindlikkust ja võimekust. Levinud märjistuskeelte konverterid on näiteks IronPDF [9], WeasyPrint [10] ja Flying Saucer [11].

Märjistuskeelte konverterid täidavad üldiselt vaid seda spetsiifilist tegevust, mistõttu need vajavad lõplikke andmeid hõlmavaid faile. Andmeid saab sisestada faili nende koodis loomise käigus. Teine valik on kasutada mallimootoreid (ingl. k *template engine*) nagu Thymeleaf¹ ja Jinja², millega on võimalik jäljendada mallipõhiseid generaatoreid, kus mallimootorid täidavad märjistuskeeles loodud mallis leiduvaid kohatäiteid andmetega ning konverterid loovad lõpliku PDF dokumendi.

Sel viisil dokumentide genereerimine vajab ikka peamiselt tarkvaraarenduses kasutatavaid oskuseid, et soovitud malli märjistuskeeltes luua.

3.1.3 Dokumentide ehitamise teegid

Dokumentide ehitamise teegid võimaldavad struktuuri loomise ja andmetega sisustamise otse koodist läbi käskude. Nende kasutus võimaldab täielikku paindlikkust struktuuri

¹ <https://www.thymeleaf.org/>

² <https://palletsprojects.com/p/jinja/>

poolelt, kuna hoiavad struktuuri loomise loogika koodi tasandil, kuid nõrgemate teekide puhul võivad puuduvad funktsionaalsused muutuda dokumendi loojale suurteks tõketeks. Joonisel 4 on näide ehitava teegi iText 7 Core abil PDF dokumendi loomisest.

```
public class DocumentGenerator {
    public static void generateDocument() throws FileNotFoundException
    {
        String destination = "./document.pdf";
        PdfWriter writer = new PdfWriter(destination);

        PdfDocument pdf = new PdfDocument(writer);
        Document document = new Document(pdf);

        Paragraph title = new Paragraph("See on pealkiri");
        title.setFontSize(20)
            .setBold()
            .setTextAlignment(TextAlignment.CENTER);
        document.add(title);

        document.add(new Paragraph("Lorem ipsum dolor sit amet,
consectetur adipiscing elit..."));

        document.close();
    }
}
```

Joonis 4. Näide iText 7 Core abil dokumendi ehitamisest

Koodipõhise loomisprotsessi tõttu vajab selliste vahendite ülesseadmine kõige rohkem manuaalset ja spetsialiseeritud arendaja tööd, kuna nii sisu kui ka struktuuri ja välimuse muutmiseks on vajalik generaatori koodi muutmine.

Ehitavate teekide seas on generaatorite tüüpidest suurim kogus tasuta valikuid. Suurem osa teeke on avatud lähtekoodiga ning võtab vastu arendusi laiemalt tarkvaraarenduse kogukonnalt.

3.2 Generaatori tüübi valimine

Generaatori tüübi valimine oleneb mitmetest faktoritest. Esimeseks faktoriks on organisatsiooni või arendaja finantsiline võimekus. Tasuta teeke ning rakendusi leidub palju, eriti avatud lähtekoodidega, mis koosnevad üldiselt mõnest aktiivsest arendajast ning loodavad ka laiema tarkvaraarendus kogukonna toele, et rakendusi uuendada ja toetada. Sellega kaasneb ebakindlus rakenduse kestvuses, kuna arendajatel puudub otsene

rahaline motivaator. Tasulised teegid ja teenused kaasnevad suurema kindlusega rakenduse jätkuvas arengus ja toetuses ning probleemide puhul on võimalik saada otsest ettevõttepoolset kliendituge.

Mallipõhiste generaatorite kasutus on üldiselt eelistatud, kuna see lihtsustab iga sammu automatiseerimise protsessis, kuid need kaasnevad üldiselt kõrgemate kasutuskuludega. Dokumentide ehitajate kasutamine pakub siiski peatükis 3.1.3 kirjeldatud paindlikkust ning sobib ilma lisakuludeta dokumentide genereerimiseks kui mallile pidevalt muudatuste tegemine ei ole nõutud. Need pakuvad ka head baasi oma vajadustele spetsiifiliselt loodud dokumente genereerivate süsteemide ülesehitamiseks.

Teekide või teenuste vahel valimisel tulevad aktuaalseteks omadusteks tehnilised võimekused. Teegid eeldavad täielikku ülesseadmist koodi tasandil ning vajavad suuremate muudatuste tegemiseks arendajate sisendit. Sellele vastanduvad pilvepõhised teenused, mis pakuvad *low-code*, ehk minimaalse arendusvajadusega, või isegi *zero-code*, ehk kasutuseks täielikult koodivabasisid lahendusi.

Teekide põhiste lahenduste juures peab arvestama sellega, et genereerimissüsteem peab üldiselt jooksva kliendi enda serverites ning vajavad seetõttu rohkem ülesseadmist ja hooldust. See võib olla positiivne külg tundlike andmete korral, kuna pilvepõhiste teenuste kasutamine eeldab nende edastamist läbi interneti teistele ettevõtetele. Teekide valikut võivad kitsendada arendajate oskused ning organisatsiooni ja selle süsteemide standardid, kuna teegid on programmeerimiskeeltele spetsiifilised.

3.3 Kasutusõigus

Dokumente genereerivate tööriistade legaalseks kasutuseks peavad ettevõtted ja organisatsioonid kindlustama, et nende kasutus on nende spetsiifilises olukorras legalne. Nende seas leidub nii tasuta kui ka tasulisi lahendusi. Tasuliste lahendustega kaasneb harilikult vastav teenusepakkuja litsents, mis võimaldab kasutust äritegevuses ilma suuremate piiranguteta, kuid siiski on kliendil vajalik tutvuda litsentsi spetsiifikaga. Litsentside omapäradest näide võib olla ReportLab PLUS litsents, mille kasutamise hind on kahekordne finantsteenustega tegelevatele ettevõtetele [12].

Tasuta tööriistade kasutusel leidub rohkem nüansse. Reeglina on tasuta genereerivad tööriistad levitatud avatud lähtekoodiga, mis võimaldab kasutajatel muuta lähtekoodi

vastamaks rohkem oma vajadustele ning avaldada modifitseeritud versioonid eraldi rakendusena. Nendega kaasnevate litsentside korral tuleb olla teadlik litsentside spetsiifikast ning nii autori kui ka kasutaja õigustest ja kohustustest enne kasutamist, kuigi suurem osa kasutavad vägagi levinud litsentse.

Järgnevalt kirjeldatakse avatud lähtekoodiga PDF generaatorite seas levinumaid litsentse, ranguse poolest kasvavas järjekorras. Litsentseid omavad nendest eelnevate omadusi, kui pole neid eraldi välja toodud.

3.3.1 MIT License

MIT (*Massachusetts Institute of Technology*) litsents on üks lihtsamaid ja levinuimaid avatud lähtekoodiga tarkvara litsentse. Litsents kuulub *permissive* litsentside kategooriasse, mis on kõige rohkemate kasutusvabadustega kategooria. Litsents lubab kasutajatel muuta lähtekoodi, jagada muudetud koodi edasi tugevama litsentsi all ja kasutada koodi nii isiklikes kui ka kommertstarkvarades. MIT litsents pakub äärmiselt palju vabadust, eeldusega vaid, et kasutatav tarkvara kuvab litsentsi omamist ja sisaldab tervet litsentsi teksti. Selle litsentsiga ei vastuta koodi autor kuidagi kasutaja ees [13]. Selle litsentsiga on loodud näiteks XDocReport [6], QuestPDF [14] ja pdfmake [15].

3.3.2 BSD License

BSD (*Berkeley Source Distribution*) litsents jaotub nelja erinevasse versiooni: nelja, kolme, kahe ja null klausliga litsents, mis erinevad ranguse poole pealt. Suurimat kasutust näeb tänapäeval kolme klausliga litsents, mis sarnaneb MIT litsentsile, kuid keelab modifitseeritud lähtekoodi reklaamimisel mainida originaalse koodi autorite nimesid ilma kirjaliku nõusolekuta [16]. Seda litsentsi kasutavad näiteks ReportLab [17] ja WeasyPrint [10].

3.3.3 Apache License 2.0

Apache License 2.0 sarnaneb laialdaselt MIT litsentsiga. Apache 2.0 litsents kohustab lähtekoodi muudatuste tegemisel nende välja toomise ning võimaldab muudetud koodi patenteerimist [18]. Seda litsentsi kasutab näiteks Apache PDFBox [19].

3.3.4 Mozilla Public License 2.0

Mozilla Public License 2.0 on nimekirja esimene *copyleft* litsents, ehk paneb koodi kasutajale rohkem kohustusi kui lihtsat viitamist. Erinevalt eelnevatest kohustab MPL 2.0

all oleva lähtekoodi muutmine ja jagamine sama litsentsi edasi kandumist, kuid vaid failidel, mis hoiustavad selle litsentsi all olevat originaalset koodi. Lisaks eemaldab MPL 2.0 patendi rikkumise korral rohkem õiguseid originaalse koodi kasutajalt kui Apache License 2.0 [20]. Seda litsentsi kasutab näiteks OpenPDF [21].

3.3.5 GNU Lesser General Public License

LGPL litsentsi kasutatakse peamiselt teekidel. See täiendab MPL 2.0 litsentsi vajadust omada sama litsentsi, kuid LGPL korral tervele teegile. Lisaks eemaldab LGPL võimaluse originaalset koodi all-litsentseerida [22]. Seda litsentsi kasutavad näiteks OpenPDF [21], TCPDF [23] ja JasperReports [8].

3.3.6 GNU Affero General Public License

AGPL litsents täiendab veelgi LGPL litsentsi. Kasutades AGPL litsentsiga teeke oma rakenduses, seega pakkudes avalikkusele teegi funktsionaalsust, isegi kaudselt, kohustab kasutajal avada terve raskenduse lähtekood. Reegel kehtib vaid avalikkusele suunatud rakenduste korral, organisatsioonisisene kasutus seda ei kohusta [24]. Eriti paistab probleem välja SaaS ettevõtete kontekstis, kus peaks AGPL litsentsi omava koodi kasutusel avaldama terve rakenduse lähtekoodi, selle tõttu on levinud soov hoiduda selle litsentsi all olevate rakenduste kasutamisest. Sellist poliitikat järgib näiteks Google, keelates AGPL all litsentseeritud koodi kasutust [25]. Seda litsentsi kasutab üks populaarsemaid PDF genereerimise teeke, iText Core [26].

3.4 Generaatorite võrdlus

Dokumentide genereerimiseks leidub paljusid teeke ja teenuseid, eristatud peamiselt eelnevalt kirjeldatud omaduste järgi. Tabelis 2 on välja toodud mõned levinud teegid ja teenused ning nende omadused.

Nimi	Funktsionaalsus	Andmete edastus	Hind	Litsents
iText Core	Käsu põhine ehitav teek (Java ja C#)	Koodist	Tasuta / Tasuline	AGPL / Kommertslitsents
Apache PDFBox	Käsu põhine ehitav teek (Java)	Koodist	Tasuta	Apache License 2.0

OpenPDF	Käsu põhine ehitav teek (Java)	Koodist	Tasuta	LGPL ja Mozilla Public License
IronPDF	HTML / XML konverter teek (C#, VB.net, F#, Java)	Koodist	Tasuline	Kommerts litsents
TCPDF	Käsu põhine ehitav teek (PHP)	Koodist	Tasuta	LGPL
pdfmake	Käsu põhine ehitav teek (JavaScript)	Koodist	Tasuta	MIT License
WeasyPrint	HTML konverter teek (Python)	Koodist	Tasuta	BSD License
Flying Saucer	XML / XHTML konverter teek (Java)	Koodist	Tasuta	LGPL
QuestPDF	Käsu põhine ehitav teek (C#)	Koodist	Tasuta / Tasuline	MIT License / Kommerts litsents
ReportLab	Käsu põhine ehitav teek (Python)	Koodist	Tasuta	BSD License
XDocReport	DOCX / ODT malli põhiselt genereeriv teek (Java)	Koodist	Tasuta	MIT License / LGPL
JasperReports	Malli disaineriga genereeriv teek	Koodist	Tasuta / Tasuline	LGPL / Kommerts litsents
PDFMonkey	HTML malli põhiselt genereeriv pilve teenus	JSON	Tasuline	Kommerts litsents
Docmosis	DOCX / ODT malli põhiselt genereeriv pilve teenus / teek (Java)	JSON / XML / koodist	Tasuline	Kommerts litsents
iText DITO	Malli disaineriga genereeriv pilve teenus	JSON	Tasuline	Kommerts litsents
Windward Hub	Malli disaineriga genereeriv pilve teenus	JSON / XML / andmebaasi seos	Tasuline	Kommerts litsents

Tabel 2. Dokumentide genereerimise teegid ja teenused ning nende omadused [5]–[11], [14], [15], [17], [19], [21], [23], [26], [27], [28].

4 Nõuete kogumine

Käesolevas peatükis analüüsib autor kliendi nõudeid rahastusplatvormi aruannete genereerimiseks ning platvormi tehnilist struktuuri ja tarkvaraarenduse häid praktikaid, eesmärgiga panna paika täpne plaan lõpliku lahenduse arenduseks.

4.1 Nõuete küsimused

Eduka tarkvaraarenduse vundamendiks on kindlalt defineeritud nõuded. Nõuete kindlustamiseks on kasulik tarkvara nõuete spetsifikatsiooni (ingl. k *Software requirements specification*) ehk SRS loomine. Selle loomise eesmärk on kindlustada arendaja ja kliendi arusaamade kattumine, et lõplik rakendus vastaks kliendi vajadustele ja oleks arenduse poolt reaalne ja jätkusuutlik. IEEE SRS soovitude järgi peaksid loomisel olema põhiküsimused järgmised [29]:

- Mida peaks rakendus tegema?
- Kuidas käitub tarkvara inimeste, süsteemi riistvara, muu riistvara ning muu tarkvaraga?
- Mis on rakenduse kiirus, kättesaadavus, reaktsiooniaeg, taastumise aeg jne?
- Mis on teisalduvuse, korrektsuse, hooldatavuse, turvalisuse jne kaalutlused?
- Kas kehtivad nõutavad standardid, rakenduskeel, andmebaasi terviklikkuse reeglid, ressursipiirangud, töökeskkonnad jne?

Järgmiste alapeatükkide jooksul proovib autor leida kliendi soovide, olemasoleva süsteemi analüüsi ja tarkvaraarenduse parimate praktikate põhjal sobivad vastused antud küsimustele.

4.2 Nõuded kliendilt

Enne ülejäänud rakenduse analüüsi tuleb leida põhiline probleem kliendilt, mille lahendamine oleks antud arendusprotsessi eesmärk. Karl Wieggers ja Joy Beatty nimetavad efektiiveteks kliendilt nõuete kogumise viisideks järgmiseid tehnikaid [30]:

- Intervjuude pidamine pakub lihtsaimat sisendi saamist, eeldab üht või mitut inimest, kellelt otse küsida sisendit planeeritud rakenduse teemadel. Head soovitusel on enda tutvustamine, intervjuu eesmärkide jagamine, küsimuste ette valmistamine, ideede pakkumine ja aktiivne kuulamine.
- Töötubade korraldamine näeb ette suurema grupiga koostöös nõuete väljatöötamist. Efektiivseks töötamiseks aitab erinevate lahendusega seotud isikute ja valdkonna spetsialistide kaasamine, kindlate eesmärkide ja reeglite loomine, ajaliste piirangute loomine ning kõigi kaasatuse kindlustamine. Töötoad võimaldavad rohkem läbimõeldud nõuete loomist, kuid vajavad olulist eeltööd ja planeerimist, et neid edukalt läbi viia.
- Fookusgruppide eesmärk on koguda erinevatelt kasutajaklassidelt sisendit. Erinevalt töötubadest on fookusgruppide peamine eesmärk kasutajate subjektiivsete arvamuste ja soovide kogumine, mitte kindlate nõuete väljatöötamine.
- Lõppkasutajate vaatlus võimaldab uurijal koguda läbi kasutaja töövoogude vaatluse sisendit rakenduse nõuetesse. Kasu tuleneb sellest, et inimestel võib olla raskuseid oma töövoogude detailide välja toomises nii ülesannete keerukuse kui ka harjumuste tõttu. See võimaldab varem kogutud puudulikkuste kinnitamist kui ka potentsiaalselt vaatleja värske perspektiivi tõttu uute lahenduste välja mõtlemist.
- Küsimustikud on lihtsaim ja odavaim moodus koguda arvamusi suurelt grupilt inimestelt. Küsimustikud on hea sisend teiste tehnikate kasutuseks, et koguda levinuimaid arvamusi ning neid teemasid tõstatada teistel viisidel. Nende puudus on sisuka info raske saamine ning lisaküsimuste võimaluse puudus ning heade ja sisukate küsimuste loomise raskus.

Kliendiks antud arendusel on TalTechi üliõpilasesindus, kelle peamine vastutav tudengielu rahastamise teemadel ja aruannete valiidsuse kinnitamisel on rahastuskomisjoni esimees. Olles peamine aruannete nõuete tundja ning loodava funktsionaalsuse kasutaja, valis autor kliendi nõuete kogumiseks intervjuu stiilis informatsiooni kogumise.

Intervjuu käigus ilmusid peamisteks nõueteks aruannete informatsiooni korrektsus, kasutajakogemuse lihtsus, maksimaalne aktiivaeg ning tulevikuarenduse ning hooldustöö lihtsus. Lisaks tuli välja lahenduse finantsiline ootus. Aruannete genereerimiseks ei ole üliõpilasesindusel arvestatud rahalisi kulusi, mis paneb piiranguid dokumente genereerivate vahendite valikus.

Dokumenditüüpi, mille jaoks sooviti genereerimist on konkursi seletuskiri, mis hoiustab peamist informatsiooni konkursi ning avalduste kohta, nende hindamistulemusi ning hindamiskomisjoni selgitusi. Seletuskirju on loodud terve konkursside olemasolu jooksul, mistõttu on nendel olemasolev struktuur, mida genereeritavad dokumendid peavad järgima (Lisa 2).

Dokumendid võivad vajada lihtsaid välimuse või struktuuri muudatusi nende kasutuse jooksul. Muudatuste soovimine võib toimuda kliendi poolt tihti, mistõttu parima kasutajakogemuse tagamiseks on vajalik lahendus, mis teeb muudatuste tegemise võimalikult lihtsaks. Siiski üliõpilasesinduse nõuetes puudus eeldus arendaja sekkumiseta muutmiseks, kuna personali seas on IT koordinaator, kelle tööülesannete seas on rahastusplatvormi arendamine ning hooldamine.

4.2.1 Kasutajalugu

Intervjuu põhjal saab kliendi kasutajakogemuse soove kokku võtta kasutajaloona: rahastuskomisjoni esimehena tahan ühe klikiga alla laadida hinnatud konkursi seletuskirja, et täita aruandvaid kohustusi.

Rahastuskomisjoni esimees vastutab seletuskirjade eest ning on rahastusplatvormil rolliga MANAGER, mistõttu peab genereerimine eeldama antud rolli. Lisaks peab rakendus kindlustama vajalike andmete olemasolu läbi konkursi hinnatud oleku, kontrollides konkursi staatust, mis peab olema GRADED. Vajalike andmete olemasolu rahastusplatvormi andmebaasis võimaldab seletuskirja genereerimist ühe klikiga.

4.3 Tehnilised nõuded

Tehniliste nõuete kogumise eesmärk on kindlustada rakenduse vastavust kliendi ootustele ning kindlustada selle hooldatavust ja vastavust laialt aktsepteeritud tarkvara arenduse printsiipidega.

Lahenduse loomisel tuleb kasuks Edsger W. Dijkstra poolt loodud põhimõte, kohustuste lahusus (ingl. k *Separation of concerns*), mille eesmärk on eraldada tarkvara erinevateks tükkiideks, mis täidavad oma eesmärgi. See põhimõte aitab hoida fookust antud ülesandel, võimaldab taaskasutust ning lihtsustab oluliselt edaspidist arendust, hoidudes ebavajalikust koodi seostusest [31].

Antud põhimõtet saab rakendada eraldades aruandeid genereeriv funktsionaalsus seletuskirja spetsiifilistest omadustest, mis võimaldab selle kasutust teiste dokumentide loomiseks. Seega lahendus vajab struktuuri, mis oleks reguleeritav igat tüüpi vajadustele. Põhimõtte põhjal tuleneb ka eeldus, et funktsionaalsus ei mõjuta kuidagi teisi toiminguid. Genereerimisprotsess ei tohi häirida muid süsteemi tegevusi ja töökindlust.

Kohustuste lahususe printsiipi järgides tuleb kaaluda ideed, kas funktsionaalsus eraldub peamisest rakendusest piisavalt, et see näeks kasu täielikust eraldatusest.

Mikroteenuste arhitektuur on lähenemine, mis jaotab laiema rakenduse funktsionaalsused väiksematesse ja iseseisvatesse moodulitesse. Selle lähenemise eesmärk on lihtsustada arendusprotsessi suurte rakenduste korral, hoida süsteemide paindlikkust ning teha pideva integratsiooni (ingl. k *Continuous integration*) ja uuenduste väljalaskmise protsessid sujuvamaks. Ressursi intensiivsete protsesside korral võimaldab see spetsiifiliste toimingute võimekuse horisontaalset skaleerimist. Lisaks pakuvad eraldatud teenused isolatsiooni probleemide eest, kindlustades, et ühe rakenduse töö lakkamine ei mõjuta teisi teenuseid. Selle lähenemisega kaasnevad ka tervikliku süsteemi keerukus ning suurem vaev hoolduse näol [32].

Dokumentide genereerimine on ülejäänud rakenduse suhtes funktsionaalsus, mis näeks palju kasu eraldusest mikroteenusesse, kuid rahastusplatvormi väikse arendusmeeskonna ja uute versioonide harva arenduse tõttu ei väljenduks mitmed mikroteenuste arhitektuuriga kaasnevad arendusprotsessidega seotud positiivsed küljed. Olles osa platvormi peamisest *back-end* süsteemist võimaldab olemasoleva andmepääsukihi ja

äriloogika ära kasutamist ning eemaldab vajaduse hooldada eraldi teenust. Väikese kasutuse ja sihtgrupi tõttu puudub vajadus võimaldada skaleerimist. See-eest kohustab see suuremat fookust tõrketöötlusel ja kindlustamisel, et selle vead ei kahjustaks ülejäänud süsteemi toimimist.

Süsteemil puuduvad andmebaasi terviklikkuse mured, kuna aruannete genereerimine ei muunda andmebaasis olevaid andmeid.

Dokumentide genereerimine ja saatmine on arusaadavalt suured toimingud, kuid rakenduse mugavaks kasutamiseks ning kasutaja tähelepanu hoidmiseks on hea tagastada tulemusi võimalikult kiiresti. Jakob Nielseni sõnul on 10 sekundit piir kasutaja tähelepanu hoidmiseks, peale mida soovib kasutaja hakata muude tegevustega oote ajal tegutsema [33]. Seega on see sobiv ülemine piir vastuse tagastamiseks.

4.4 Olemasoleva rakenduse nõuded

Rahastusplatvormile arenduste tegemisel peab arvestama olemasoleva rakenduse struktuuriga, mis koosneb kolmest osast [1], [2]:

- Andmebaasiks on kasutusel PostgreSQL¹ 11.4.
- Rakendusliides on Spring Boot² 2.2 raamistikku kasutatav Java 11 rakendus.
- Kasutajaliides on Vue.js³ 2 raamistiku põhine *Single Page Application*.

Dokumentide genereerimine on ressursiintensiivne toiming, mistõttu ei tohiks protsess toimuda kasutajaliidese tasandil. Valik funktsionaalsust mitte eraldada mikroteenusesse kohustab seega genereerimise lisamist rakendusliidesesse, mis eeldab lahenduse loomist Java programmeerimiskeeles.

¹ <https://www.postgresql.org/>

² <https://spring.io/projects/spring-boot>

³ <https://vuejs.org/>

5 Arendatud lahendus

Antud peatükis kirjeldab autor rahastusplatvormi aruannete genereerimise lahenduse loomiseks tehtud valikuid põhinedes peatükis 4 paika pandud nõuetele, rahastusplatvormile tehtud muudatusi ning lahenduse tööprotsessi.

5.1 Generaatori valik

Dokumente genereeriva tööriista valikus tulid peamiseks olulisteks nõueteks tööriista hind ning vastavus Java programmeerimiskeeles loodud platvormi rakendusliideselega.

Tööriistaks sai valitud OpenPDF, mis on avatud lähtekoodiga PDF failide loomiseks ja muutmiseks loodud dokumenti ehitav Java teek. OpenPDF on Git *fork* populaarseimast Java genereerimise teegist, iText Core [21]. Hargnemise eesmärk oli hoida LGPL litsentsi, mis sai iText 5 saabumisel asendatud AGPL litsentsiga [34], mille kasutus kaasaks arendajatele kohustusi, mis on kirjeldatud peatükis 3.3.6.

OpenPDF on aktiivselt arendatud teek, mis kindlustab selle ajakohasuse ning põhinemine laialt levinud iText teegil lihtsustab kasutuskogemust, võimaldades iText laia dokumentatsiooni ning kasutajaskonna toe ära kasutamist.

5.1.1 Mallipõhised generaatorid

Lihtsaimat kasutajapoolset dokumendi välimuse ja struktuuri muutmist võimaldavad mallipõhised generaatorid, kuid selle lahenduse uurimisel ilmnes puudus tasuta DOCX ja ODT mallipõhistes generaatorites, mille kasutamine vajaks vaid olemasoleva dokumendi malli ümbersättimist. Puuduseks on peatükis 2.3 kirjeldatud sektsioonide kordamise võimekus, mis puudub valikul olevates tasuta mallipõhistes generaatorites, kuigi need on leitavad paljudes kommertstarkvarades, nagu näiteks Docmosis, mille kasutus oleks üliõpilasesindusele liiga kulukas.

Mallimootori ja märgistuskeele konverteri põhjal mallipõhise generaatori valik võimaldab soovitud sektsioonide kordamist, kuid ei eemaldaks muudatuste tegemisel märgistuskeele, CSS ja mallimootori teadmistega arendaja sekkumise vajadust. DOCX

muutmine märgistuskeele formaati lahendaks seda probleemi, kuid muutmise jooksul kaob tihti eraldatud lehekülgedeks küljendatus ning väljund võib olla vigane struktuuri poole pealt, mis raskendab oluliselt automatiseerimist.

Lisaks pakub lihtne ning tulemust kirjeldav koodipõhine lahendus lihtsamat muutmise ning dokumendi ehitamise protsessi kui pikad märgistuskeelte failid.

5.2 Lahenduse struktuur

Käsupõhised dokumendi ehitajad pakuvad arendajale palju vabadust dokumendi loomisel, mis kaasneb vajadusega iga elemendi puhul täpsustada kõiki nende omadusi, mis põhjustab pikka ning korduvat koodi. Muudatuste läbiviimise ning uute dokumentide loomise lihtsustamiseks sai rakendusliidese lahendus loodud kahes osas:

- Dokumendi ehitaja ReportBuilder.
- Ehitaja rakendamine spetsiifilise dokumendi loomiseks.

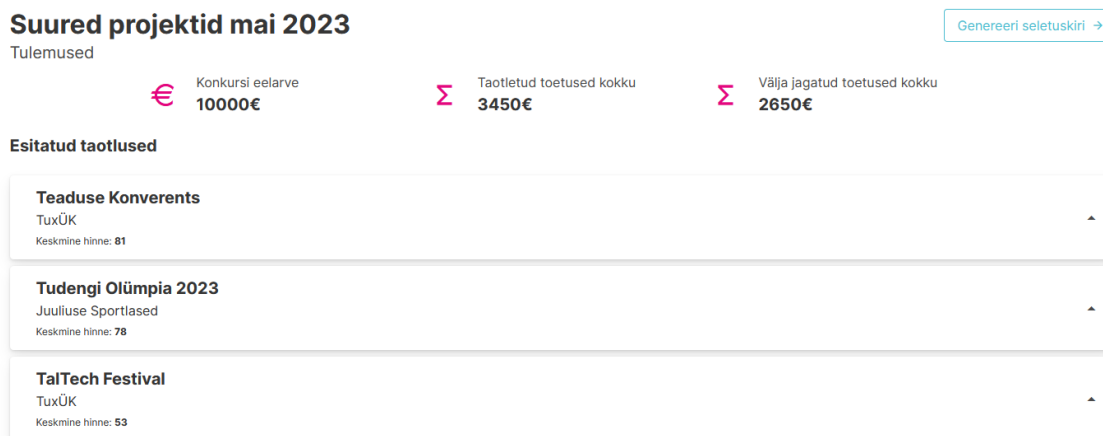
Dokumendi ehitav klass ReportBuilder järgib üht niinimetatud *Gang of Four* disainimustrite seast, ehitaja (*Builder*) mustrit. Raamatus “Design Patterns: Elements of Reusable Object-Oriented Software” kirjeldatakse ehitaja mustrit kui viisi, kuidas ehitada keerulisi objekte tükikaupa läbi abstraktse liidese, peites sisemist töötamist kasutaja eest. Seda põhimõtet järgides on klassi eesmärk võimaldada dokumendi ehitamist elementide ning nende omaduste põhjal ilma vajaduseta teada detaile teegi töötamise kohta [35].

Dokumentide ehitajas puudub spetsiifiline struktuuri ning välimuse spetsiifika, mis võimaldab sellel põhinevate objektide ümber seadistamist mis iganes dokumendi loomiseks. Dokumentide üldiselt ühtlase disaini tõttu piisab ehitaja poolt loodavate elementide ühekordsest konfigureerimisest. Seda põhimõtet järgides oleks ehitajat võimalik taaskasutada nii rahastusplatvormi siseselt, et teisi dokumente luua, kui ka teistes süsteemides.

5.3 Muudatused kasutajaliideses

Vastavalt nõudega käivitada seletuskirja genereerimine ühe klikiga olid muudatused kasutajaliideses minimaalsed. Konkursi tulemuste vaatele sai lisatud nupp tekstiga

„Genereeri seletuskiri“, mis esineb kasutajaliideses vaid siis, kui hinnangud on tehtud avalikuks ja konkursi staatus on GRADED ning kasutaja omab rolli MANAGER. Joonisel 5 on näha konkursi tulemuste vaade testandmetega, kui antud eeldused on täidetud.



Joonis 5. Konkursi tulemuste vaade.

Nupu vajutusel saadab kasutajaliides HTTP päringu rakendusliidesele ning edukal jooksmisel avaneb kasutajale uus aken seletuskirjaga. Päringu tegemisel edastatakse contestId, mis täpsustab konkursi mille seletuskirja soovitakse, ning kasutaja autentimiseks vajalik JWT¹ token. Vigade ilmnmisel antakse kasutajale sellest teada veateate näol. Tabelis 3 kirjeldatakse tehtavat päringut.

URI	Meetod	Tagastus	Kirjeldus
/result/{contestId}/generateReport	GET	application/pdf	Tagastab kasutajale genereeritud dokumendi.

Tabel 3. Dokumendi genereerimise päring.

5.4 Muudatused rakendusliideses

Rahastusplatvormi rakendusliides järgib kolmekihilist API struktuuri, mis koosneb järgmistest osadest [1]:

- Esitluskiht, ehk *controller*.

¹ <https://jwt.io/>

- Äriloogika kiht, ehk *service*.
- Andmepääsu kiht, ehk *repository*.

Kasutajaliidese tehtud päring, kirjeldatud tabelis 3, võetakse vastu esitluskihis vastava otspunkti poolt, kus kontrollitakse päringut tegeva kasutaja Auth0¹ rolli, kindlustamaks päringu tegija seostust üliõpilasesindusega, seejärel saadetakse päring edasi äriloogika kihile.

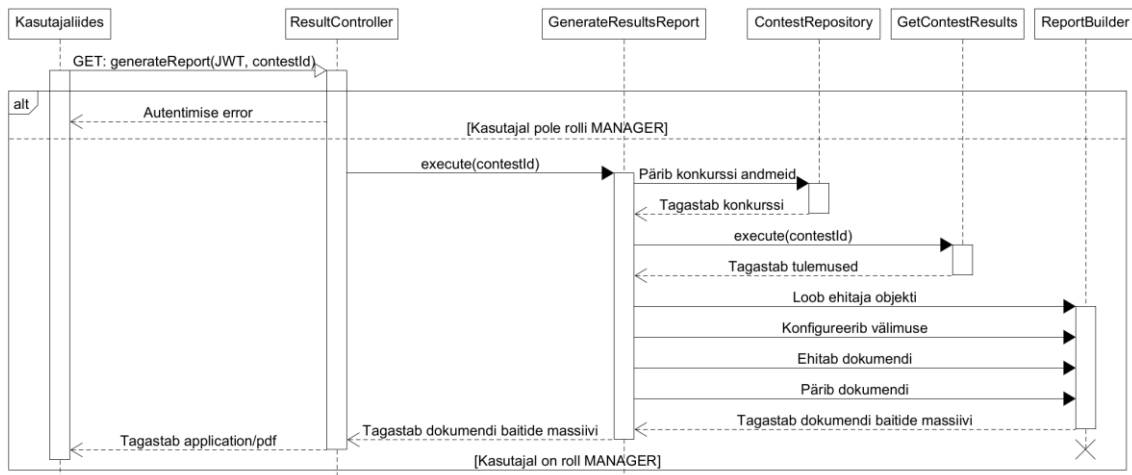
Äriloogika kiht on rahastusplatvormil ehitatud üles kasutajalookeskse arhitektuuri põhimõtteid järgides, kus iga UseCase klass täidab üht spetsiifilist eesmärki [1].

Äriloogika kihi klass `GenerateResultsReport` kogub konkursiga seotud informatsiooni andmepääsukihi liideselt `ContestRepository` ning taaskasutab taotluste tulemustega vajalikud andmed olemasolevalt UseCase klassilt. Seejärel luuakse dokumendi ehitaja `ReportBuilder` objekt, sätestatakse omadused ning lisatakse ehitaja käskude kaudu vajalikud elemendid koos andmetega. Valminud dokumendi näide on leitav Lisas 3.

Dokumentide korduva kasutamise puhul oleks parim valik kasutajale edastamiseks faili ühe korra loomine ja hoiustamine ning selle asukoha edastamine kasutajatele, et hoida kokku korduvatelt genereerimiskuludelt. Antud nõuete puhul pole see vajalik, kuna seletuskiri saab üliõpilasesinduse protsesside järgi loodud vaid korra, alla laetud, ning edastatud digiallkirjaga üliõpilasesinduse failihaldussüsteemi. Funktsionaalsust kasutatakse ühe konkursi jaoks korduvalt vaid siis, kui andmetele on tehtud muudatusi, mis eeldaks igal juhul uut seletuskirja genereerimist. Nii hoitakse kokku ka failide hoiustamise kuludelt.

Seletuskirjade hoiustamise vajaduse puudumise tõttu luuakse dokument baitide massiivina (ingl. k *byte array*), mis tagastatakse valmidusel esitluskihile ning edastatakse kasutajale päringu vastusena. Joonisel 6 on kujutatud rakendusliidese genereerimise protsess diagrammina.

¹ <https://auth0.com/>



Joonis 6. Rakendusliideses seletuskirja genereerimise protsessi diagramm.

5.5 Dokumendi ehitaja kasutamine

Peatükis 5.2 sai kirjeldatud põhimõtted, mida järgides sai dokumentide ehitaja loodud, milleks olid taaskasutuse võimekus ning lihtne ja loetav kood.

Dokumendi omaduste sätestamiseks kasutatakse käskude aheldamist (ingl. *k method chaining*), mille tõttu on sätestamine lühike ning loetav. Elementide levinuimad omadused on seatud vaikimisi, mistõttu pole kasutajal vaja kõiki omadusi manuaalselt sättida. Dokumendi omaduste sätestamise näide on välja toodud joonisel 7.

```

this.report = new ReportBuilder()
    .addDesignImage(LOGO_MAIN_PATH, 20, 80, 710)
    .addDesignImage(LOGO_SIDE_PATH, 50, 550, 660)
    .setTextFont(CALIBRI_PATH, 11)
    .setBoldTextFont(CALIBRI_BOLD_PATH, 11)
    .setTitleFont(CALIBRI_BOLD_PATH, 12)
    .setSubtitle1Font(CALIBRI_BOLD_PATH, 11);

```

Joonis 7. ReportBuilder loomise ja dokumendi omaduste konfigureerimise näide.

Elemente dokumenti lisavad käsud kirjeldavad vaid elemente ning nende üldiseid omadusi, kuna omaduste sätestamine on eelnevalt tehtud. Seletuskirja esimest lehte loov kood on joonisel 8.

```

private void addFrontPage(Contest contest,
List<ContestReportApplicationDto> applications) {
    report.addTitle(getTitle(contest));
    report.newLine();
    report.addSubtitle1("I.      ÜLDSÄTTED");
    report.newLine();
    report.addText(getIntroText(contest, applications));
    report.newLine();
    report.addUnderlinedText(getCriteriaTitle(contest));
    report.addBulletList(getCriteria(contest));
    report.newLine();
    report.addUnderlinedText("Komisjoni koosseis:");
    report.addOrderedList(getCommitteeMembers(contest));
}

```

Joonis 8. ReportBuilder kasutamise näide esilehe loomiseks.

6 Tulemused ja analüüs

Käesolevas peatükis antakse ülevaade lõputöö raames loodud lahenduse vastavusest nõutele ning vajalikest edasiarendustest.

6.1 Tulemuste valideerimine

Korrektsete tulemuste valideerimiseks sai genereerimise funktsionaalsus läbi tehtud erinevate konkursi tüüpidega ning visuaalse uurimise põhjal kindlustatud andmete ja struktuuri korrektsus. Lahendus sai kontrollitud ka ebatavaliste andmetega, nagu pikkade sõnedega ning *null* väärtusega, kus neid võib platvormil tekkida.

Lahendus sai valideeritud rahastuskomisjoni esimehe poolt arenduskeskkonna demonstratsiooni kaudu, kus jälgiti kasutajaloo täituvust ning valminud dokumendi struktuuri ja andmete täpsust. Demonstratsiooni jooksul oldi lahendusega rahul, pakkudes välja vaid väikseid visuaalseid muudatusi genereeritud dokumendile, mis viidi järgnevalt ellu.

Autori arvates on lahendus vägagi adekvaatne ning pakub tulevikus muudatuste tegemiseks ning teiste dokumentide genereerimiseks sobivat lahendust. Siiski oleks sarnast lahendust olnud võimalik luua ka märgistuskeele konverteri ja mallimootori abil, mis oleks lihtsustanud oluliselt arendusprotsessi, kuid teoreetiliselt raskendanud eelnevalt mainitud muudatuste tegemist.

6.1.1 Nõuetele vastavus

Lahendus genereerib kliendi nõuetele vastavalt ühe nupu vajutusega kasutajale valmis ning korrektse konkursi seletuskirja. Dokumentide ehitaja struktuuriga on võimaldatud taaskasutus teiste dokumentide loomiseks ning ehitamine ja muutmine on käskude ja dokumendi sätestamisega tehtud üliõpilasesinduse IT koordinaatorile lihtsaks.

Funktsionaalsuse kättesaadavus on seotud platvormi kättesaadavusega ning puudub vajadus eraldi teenust hooldada. Siiski ei sega potentsiaalselt pikad dokumentide genereerimise ajad teisi toiminguid, kuna platvorm jooksub Spring Boot rakendusliidest

Tomcat¹ serveril. Selle abil töödeldakse toimingud eraldi lõimedel (ingl. k *thread*), mis võimaldab toimingute üheaegset ja eraldiseisvat töötlust [36].

Dokumendi genereerimise reaalsel kiirust pole olnud võimalik mõõta, kuna lahendus pole tarbekeskonda (ingl. k *production environment*) juurutatud. Siiski, lokaalselt jooksva arenduskeskkonna korral testiti korduvalt erinevate konkursitüüpide ja näiteandmetega seletuskirjade genereerimist ning Firefox brauserisisese võrgumonitori² järgi ei võtnud ükski päring aega üle 2 sekundi. Tarbekeskonnas suureneb kahtlemata genereerimise aeg, kuid lokaalselt sai kindlustatud, et dokumendi genereerimise protsess ise ei võta suurt kogust aega.

6.2 Edasiarendus

Töö käigus sai loodud üliõpilasesinduse nõuetele vastav lahendus, kuid ei jõutud selle tarbekeskonda juurutamiseni.

Ehitaja funktsionaalsuste arendusel alustati elementide ja omadustega, mis olid seletuskirja genereerimiseks ja tulevaseks modifitseerimiseks vajalikud. Loodud lahendusel leidub juba sarnase ehitusega dokumentide genereerimiseks piisavalt paindlikkust, kuid kõigi peatükis 2.2 mainitud elementide loomiseks ja omaduste muutmiseks vajab lahendus veel edasiarendust.

Juurutamiseks valmis oleva koodi eeldus on ka põhjalik testimine, mida ei jõutud lõputöö raames piisavalt sisukalt läbi viia. Seega on vajalik edasiarendus suurema koguse ühik- ja integratsioonitestide loomine ning edasine manuaalne testimine äärmusjuhtumitega (ingl. k *edge case*).

Ehitaja loomine fokuseeris selle paindlikkusele erinevate dokumentide tegemiseks ning selle eraldust ülejäänud platvormi funktsionaalsusest. See lähenemine võimaldaks ka ehitaja kasutust erinevatel platvormidel, mille võimaldamiseks saaks lahendust arendada edasi eraldiseisvaks avatud lähtekoodiga teegiks. See eeldaks eelnevalt mainitud rohkemat funktsionaalsust ja testimist ning failisest dokumenteerimist kasutades

¹ <https://tomcat.apache.org/>

² https://firefox-source-docs.mozilla.org/devtools-user/network_monitor/

Javadoc¹ vahendit. Ehitaja oleks vaja eraldada platvormist sellele sobiva struktuuriga eraldiseisvasse projekti ning genereerimist JAR faili. Valmisoleva ja väärtust pakkuva teegi jagamist maailmaga võimaldaks näiteks Maven Central² [37].

¹ <https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html>

² <https://central.sonatype.com/>

7 Kokkuvõte

Töö eesmärgiks oli arendada TalTech üliõpilasesinduse rahastusplatvormile automaatne aruannete genereerimise funktsionaalsus, mis kiirendaks oluliselt üliõpilasesinduse rahastuskomisjoni esimehe töökohustusi, luues konkursside seletuskirju kiiresti ja mugavalt.

Lahenduse planeerimiseks uuris autor praegu kasutusel olevaid protsesse ning dokumentide elemente ja seletuskirja spetsiifikat, mille abil panna paika probleem, millele lahendust otsitakse, ning leida dokumentide omadusi, mille põhjal valida tööriista, mis aitaks nende genereerimisega.

Võimalike tööriistade kaardistamiseks analüüsis autor erinevaid tüüpe PDF dokumente automaatselt genereerivaid teenuseid ja teeke, tuues välja nende kasutuskogemusega seotud omadusi ning erinevaid faktoreid, mis võivad mõjutada tööriista valikut.

Lahenduse jaoks kindlate nõuete paika panekuks pidas autor intervjuud rahastuskomisjoni esimehega, kogudes peamiseid nõudeid, mis olid vajalikud lõpliku lahenduse loomiseks. Lisaks sai uuritud erinevaid tarkvaraarenduse põhimõtteid ja tavasid, mis võivad mõjutada lahenduse struktuuri, ning rahastusplatvormi olemasolevaid omadusi.

Kirjeldatud said lahendusega seoses tehtud valikud, lahenduse struktuur ja omapärad ning olemasoleva lahenduse tööprotsess ja platvormile sisse viidud muudatused.

Lõpuks kirjeldas autor lahenduse valideerimisprotsessi, vastavust nõuetele ning nii vajalikke kui ka töö skoobi väliseid võimalikke edasiarendusi.

Töö käigus sai loodud rahastuskomisjoni esimehe nõuetele vastav ja eesmärki täitev lahendus, kuigi ei jõutud lahendus juurutada tarbepoolsele. Loodud lahendus tehti multifunktsionaalseks, et võimaldada selle kasutust ka teiste dokumentide loomiseks, kuid täieliku potentsiaali täitmiseks vajab see veel täiendavat arendust.

Kasutatud kirjandus

- [1] K. G. Mänd, *TalTech üliõpilasesinduse veebipõhise rahastusplatvormi rakendusliidese arendus järgides kasutuslookeskse arhitektuuri põhimõtteid*, Tallinn, 2019.
- [2] R. Lume, *TalTech üliõpilasesinduse rahastusplatvormi arendus jätkusuutliku tarkvaraarenduse põhimõtteid järgides*, Tallinn, 2020.
- [3] B. Stegner, „What Is a PDF File and Why Do We Still Rely on Them?“, 13. 7. 2017. [Võrgumaterjal]. Available: <https://www.makeuseof.com/tag/what-is-pdf-file/>. [Kasutatud 7. 4. 2023].
- [4] „Docmosis Example Templates“, [Võrgumaterjal]. Available: <https://resources.docmosis.com/content/content/example-templates>. [Kasutatud 12. 4. 2023].
- [5] „Docmosis“, [Võrgumaterjal]. Available: <https://www.docmosis.com/>. [Kasutatud 21. 3. 2023].
- [6] „XDocReport GitHub“, [Võrgumaterjal]. Available: <https://github.com/opensagres/xdocreport>. [Kasutatud 21. 3. 2023].
- [7] „iText DITO“, [Võrgumaterjal]. Available: <https://itextpdf.com/products/itext-dito>. [Kasutatud 21. 3. 2023].
- [8] „JasperReports Library“, [Võrgumaterjal]. Available: <https://community.jaspersoft.com/project/jasperreports-library>. [Kasutatud 21. 3. 2023].
- [9] „IronPDF“, [Võrgumaterjal]. Available: <https://ironpdf.com/>. [Kasutatud 21. 3. 2023].
- [10] „WeasyPrint“, [Võrgumaterjal]. Available: <https://weasyprint.org/>. [Kasutatud 21. 3. 2023].
- [11] „Flying Saucer GitHub“, [Võrgumaterjal]. Available: <https://github.com/flyingsaucerproject/flyingsaucer>. [Kasutatud 21. 3. 2023].
- [12] „ReportLab PLUS Pricing“, [Võrgumaterjal]. Available: <https://www.reportlab.com/pricing/>. [Kasutatud 8. 4. 2023].
- [13] „The MIT License“, [Võrgumaterjal]. Available: <https://opensource.org/license/mit/>. [Kasutatud 14. 4. 2023].

- [14] „QuestPDF,“ [Võrgumaterjal]. Available: <https://www.questpdf.com/>. [Kasutatud 21. 3. 2023].
- [15] „pdfmake,“ [Võrgumaterjal]. Available: <http://pdfmake.org>. [Kasutatud 21. 3. 2023].
- [16] „BSD Licenses,“ [Võrgumaterjal]. Available: <https://encyclopedia.pub/entry/34847>. [Kasutatud 14. 4. 2023].
- [17] „ReportLab,“ [Võrgumaterjal]. Available: <https://www.reportlab.com/>. [Kasutatud 20. 3. 2023].
- [18] „Apache License, Version 2.0,“ [Võrgumaterjal]. Available: <https://www.apache.org/licenses/LICENSE-2.0>. [Kasutatud 14. 4. 2023].
- [19] „Apache PDFBox,“ [Võrgumaterjal]. Available: <https://pdfbox.apache.org/>. [Kasutatud 13. 4. 2023].
- [20] „Mozilla Public License,“ [Võrgumaterjal]. Available: <https://www.mozilla.org/en-US/MPL/>. [Kasutatud 14. 4. 2023].
- [21] „OpenPDF GitHub,“ [Võrgumaterjal]. Available: <https://github.com/LibrePDF/OpenPDF>. [Kasutatud 21. 3. 2023].
- [22] „GNU Lesser General Public License,“ [Võrgumaterjal]. Available: <https://www.gnu.org/licenses/lgpl-3.0.en.html>. [Kasutatud 14. 4. 2023].
- [23] „TCPDF,“ [Võrgumaterjal]. Available: <https://tcpdf.org/>. [Kasutatud 22. 3. 2023].
- [24] „GNU Affero General Public License,“ [Võrgumaterjal]. Available: <https://www.gnu.org/licenses/agpl-3.0.html>. [Kasutatud 15. 4. 2023].
- [25] „AGPL Policy,“ [Võrgumaterjal]. Available: <https://opensource.google/documentation/reference/using/agpl-policy?ref=fossa.com>. [Kasutatud 15. 4. 2023].
- [26] „iText Core,“ [Võrgumaterjal]. Available: <https://itextpdf.com/products/itext-7/itext-7-core>. [Kasutatud 19. 3. 2023].
- [27] „PDFMonkey,“ [Võrgumaterjal]. Available: <https://www.pdfmonkey.io/>. [Kasutatud 25. 4. 2023].
- [28] „Windward Hub,“ [Võrgumaterjal]. Available: <https://hub-guide.windwardstudios.com/>. [Kasutatud 24. 4. 2023].
- [29] IEEE Computer Society, „IEEE Recommended Practice for Software Requirements Specifications,“ *IEEE*, 1998.

- [30] K. Wiegers ja J. Beatty, *Software Requirements*, Microsoft Press, 2013.
- [31] E. W. Dijkstra, „Selected Writings on Computing: A personal Perspective,“ *On the Role of Scientific Thought*, New York, NY, Springer, 1982, pp. 60-66.
- [32] „Microservice architecture style,“ [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices>. [Kasutatud 29. 4. 2023].
- [33] J. Nielsen, *Usability Engineering*, Morgan Kaufmann, 1993.
- [34] „Release iText 5.0.0,“ [Võrgumaterjal]. Available: <https://kb.itextpdf.com/home/it5kb/releases/release-itext-5-0-0>. [Kasutatud 24. 3. 2023].
- [35] E. Gamma, R. Helm, R. Johnson ja J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional, 1994.
- [36] M. Mooney, „Key metrics for monitoring Tomcat,“ 10. 12. 2018. [Võrgumaterjal]. Available: <https://www.datadoghq.com/blog/tomcat-architecture-and-performance/>. [Kasutatud 4. 5. 2023].
- [37] S. Colebourne, „Best Practices for Designing and Implementing a Library in Java,“ [Võrgumaterjal]. Available: <https://www.oracle.com/corporate/features/library-in-java-best-practices.html>. [Kasutatud 2. 5. 2023].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Robert Pärnpuu

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „TalTech üliõpilasesinduse rahastusplatvormi aruannete genereerimise funktsionaalsuse arendus“, mille juhendaja on Tarvo Treier
 - 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.

17.05.2023

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

Lisa 2 – Käsitsi loodud seletuskirja näidis



2023. AASTA MAI SUURTE PROJEKTIDE RAHASTUSKOMISJONI SELETUSKIRI

I. ÜLDSÄTTED

Käesolev seletuskiri on koostatud 1. mail 2023 TalTechi rahastuskomisjoni esimehe Peeter Pärna poolt tuginedes teiste komisjoni liikmete tagasisidele ja koosoleku jooksul taotlejate esitatud küsimuste vastustele. Seletuskiri on kooskõlas Tudengielu rahastamise eeskirjaga. Komisjoni koosolek toimus 28. aprillil 2023 MS Teamsis. Konkursile laekus kokku 3 taotlust. Konkursi eelarve on 10 000 eurot.

Väikeste projektide toetuste jagamisel lähtus komisjon järgmistest kriteeriumitest:

- TalTechi tudengite kaasatus projektimeeskonda
- Üliõpilaskonna ja ülikooli nähtavus ühiskonnas
- Lisandväärtuse pakkumine TalTechi tudengkonnale
- Projekti tegevuse läbimõeldus ja teostatavus
- Projekti eelarve optimeeritus ja põhjendatus

Komisjoni koosseis:

1. Peeter Pärn – esimees
2. Heli Kopter
3. Mari Musikas

II. TALTECH ÜLIÕPILASESINDUSE TOETUS SUURTELE PROJEKTIDELE

Projektidele eraldatud toetussummad (eurodes):

Jrk. nr	Projekti nimi	Taotleja	Keskmine hinne	Taotletav summa	Eraldatud summa
1	Teaduse konverents	TuxÜK	81	850 €	850 €
2	Tudengi Olümpia 2023	Juulise Sportlased	78	1000 €	800 €
3	TalTech Festival	TuxÜK	53	1600 €	1000 €
Kokku				3450 €	2650 €

III. TOETUSTE JAGAMISE/MITTEJAGAMISE SELGITUSED

1. Teaduse Konverents

Teaduse Konverents on hästi läbi mõeldud üritus koos kogenenud korraldusmeeskonnaga. Ürituse eesmärgid ülikoolis teadusega tegelevate institutsioonide turundamiseks tudengitele on õilis eesmärk ning väga toetatud. Eelarves probleeme pole ning hindamiskomisjon toetab projekti täies mahus.

Kululiigid	Taotlev summa	Eraldatud summa
Saali rent	200	200
Esinejate kingitused	50	50
Toitlustus	500	500
Turundus	100	100
Kokku	850	850

Komisjon rahuldab taotluse täielikus mahus 850 eurot.

Taotletav: 850 eurot

Eraldatud: 850 eurot

2. Tudengi Olümpia 2023

Projekti idee on hea ning võib arendada ülikooli sportlaste kogukonda. Taotlus on põhjalik ja projekt on tudengielu maastiku arenguks vajalik. Eelarve planeerimisel leidis hindamiskomisjon kohati auke, mis tõttu projekti toetatakse vaid osalises mahus.

Kululiigid	Taotlev summa	Eraldatud summa
Sporditarvikute rent	300	300
Toit ja jook	450	300
Helitehnika rent	200	200
Võistjelite numbrite print	50	0
Kokku	1000	800

Komisjon rahuldab taotluse mittetäielikus mahus 800 eurot.

Taotletav: 1000 eurot

Eraldatud: 850 eurot

3. TalTech Festival

Tudengiulu rahastuskomisjon leidis, et ürituse taotlus oli kohati puudulik, kuid siiski TalTech Festival edendaks oluliselt tudengiulu ülikoolis. Projektimeeskond on oma kompetentsi tõestanud paljudes varasemates üritustes. Komisjon toetab üritust osalises mahus.

Kululiigid	Taotlev summa	Eraldatud summa
Esinejate sissemaksed	500	200
Helitehnika rent	300	300
Baari sissemakse	300	0
Saali rent + ehitus	500	500
Kokku	1600	1000

Komisjon rahuldab taotluse mittetäielikus mahus 1000 eurot.

Taotletav: 1600 eurot

Eraldatud: 1000 eurot

IV. LÖPPSÄTTED

Käesolev seletuskiri on edastatud üliõpilasesinduse juhatusele kinnitamiseks 1. mail 2023.

Peeter Pärn
rahastuskomisjoni esimees
peeter.pärn@tipikas.ee
+372 51234567
/allkirjastatud digitaalselt/

Lisa 3 – Genereeritud seletuskirja näidis



2023. AASTA MAI SUURTE PROJEKTIDE RAHASTUSKOMISJONI SELETUSKIRI

I. ÜLDSÄTTED

Käesolev seletuskiri on koostatud 9. mai 2023 TalTechi rahastuskomisjoni esimehe Tux Admin poolt tuginedes teiste komisjoni liikmete tagasisidele ja koosoleku jooksul taotlejate esitatud küsimuste vastustele. Seletuskiri on kooskõlas Tudengielu rahastamise eeskirjaga. Komisjoni koosolek toimus 10. mai 2023. Konkursile laekus kokku 3 taotlust. Konkursi eelarve on 10000 eurot.

Suurte projektide toetuste jagamisel lähtus komisjon järgmistest kriteeriumitest:

- TalTechi tudengite kaasatus projektimeeskonda
- Üliõpilaskonna ja ülikooli nähtavus ühiskonnas
- Lisandväärtuse pakkumine TalTechi tudengkonnale
- Projekti tegevuse läbimõeldus ja teostatavus
- Projekti eelarve optimeeritus ja põhjendatus

Komisjoni koosseis:

1. Tux Committee

II. TALTECH ÜLIÕPILASESINDUSE TOETUS SUURTE PROJEKTIDE KONKURSI RAAMES

Eraldatud toetussummad (eurodes):

Jrk. nr	Projekti nimi	Taotleja	Keskmine hinne	Taotletav summa	Eraldatud summa
1	Teaduse Konverents	TuxÜK	81	850 €	850 €
2	Tudengi Olümpia 2023	Juuliuise Sportlased	78	1000 €	800 €
3	TalTech Festival	TuxÜK	53	1600 €	1000 €
Kokku				3450 €	2650 €

III. TOETUSTE JAGAMISE/MITTEJAGAMISE SELGITUSED

1. Teaduse Konverents

Teaduse Konverents on hästi läbi mõeldud üritus koos kogenenud korraldusmeeskonnaga. Ürituse eesmärgid ülikoolis teadusega tegelevate institutsioonide turundamiseks tudengitele on õilis eesmärk ning väga toetatud. Eelarves probleeme pole ning hindamiskomisjon toetab projekti täies mahus.

Kululiigid	Taotlev summa	Eraldatud summa
Saali rent	200 €	200 €
Esinejate kingitused	50 €	50 €
Toitlustus	500 €	500 €
Turundus	100 €	100 €
Kokku	850 €	850 €

Komisjon rahuldab taotluse täielikus mahus 850 eurot.

Taotletav: 850 eurot

Eraldatud: 850 eurot

2. Tudengi Olümpia 2023

Projekti idee on hea ning võib arendada ülikooli sportlaste kogukonda. Taotlus on põhjalik ja projekt on tudengielu maastiku arenguks vajalik. Eelarve planeerimisel leidis hindamiskomisjon kohati auke, mis tõttu projekti toetatakse vaid osalises mahus.

Kululiigid	Taotlev summa	Eraldatud summa
Sportitarvikute rent	300 €	300 €
Toit ja jook	450 €	300 €
Helitehnika rent	200 €	200 €
Võistlejate numbrite print	50 €	0 €
Kokku	1000 €	800 €

Komisjon rahuldab taotluse mittetäielikus mahus 800 eurot.

Taotletav: 1000 eurot

Eraldatud: 800 eurot

3. TalTech Festival

Tudengielu rahastuskomisjon leidis, et ürituse taotlus oli kohati puudulik, kuid siiski TalTech Festival edendaks oluliselt tudengielu ülikoolis. Projektimeeskond on oma kompetentsi tõestanud paljudes varasemates üritustes. Komisjon toetab üritust osalises mahus.

Kululiigid	Taotlev summa	Eraldatud summa
Esinejate sissemaksed	500 €	200 €
Helitehnika rent	300 €	300 €
Baari sissemaksed	300 €	0 €
Saali rent + ehitus	500 €	500 €
Kokku	1600 €	1000 €

Komisjon rahuldab taotluse mittetäielikus mahus 1000 eurot.

Taotletav: 1600 eurot

Eraldatud: 1000 eurot

IV. LÕPPSÄTTED

Käesolev seletuskiri on edastatud üliõpilasesinduse juhatusele kinnitamiseks 12. mai 2023.

Tux Admin
rahastuskomisjoni esimees
tux-admin@ituk.ee