

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Tarvo Tali 210748IAIB  
Karl-Gustav Rütli 210746IAIB  
Deniel Konstantinov 206505IAIB

## **INFOSÜSTEEMIDE LOOMINE AMETIÜHINGUTELE**

Juhendaja: Margarita Matson  
PhD

Tallinn 2023

# **Autorideklaratsioon**

Kinnitame, et oleme 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.

Autorid: Tarvo Tali, Karl-Gustav Rüütli, Deniel Konstantinov

23.05.2023

# **Annotatsioon**

## **Infosüsteemide loomine ametiühingutele**

Käesoleva bakalaureusetöö eesmärgiks on ametiühingute infosüsteemi väljatöötamine Eesti Raudteelaste Ametiühingu (ERAÜ) näitel. Infosüsteem jaguneb kolmeks põhikomponendiks: koduleht, töötajate veebirakendus ja liikmete Androidi rakendus. Koduleht annab ülevaate ametiühingust, võimaldab kasutajatel lugeda uudiseid, esitada küsimusi ja liituda ametiühinguga. Töötajate veebirakendus võimaldab hallata uudiseid, liikmeid ja töötajaid ning koostada aruandeid. Android rakendus võimaldab liikmetel esitada oma liikmekaarti elektrooniliselt, ning võimaldab vaadata ja muuta isiklikke andmeid ja kontaktinfot.

Selle lõputöö käigus arendatakse esirakendusi kasutades ReactJS raamistikku, JavaScripti ja Bootstrap-i. Kodulehele luuakse PostgreSQL andmebaas, mis sisaldab uudiseid, küsimusi ja vastuseid ning liikmetaotlusi. Töötaja veebirakenduse jaoks luuakse eraldi PostgreSQL andmebaas isikuandmete jaoks ning olemasolevad andmed migreeritakse uude andmebaasi. Tagarakendused kasutavad Spring Boot-i ja Java versiooni 17. Android rakenduse arendamiseks kasutatakse Android Studiot ja Java versiooni 8.

Selle töö tulemusena valmib ERAÜ vajadustele vastav uuenduslik koduleht, töötajate veebirakendus ja liikmete Android rakendus, mis võimaldab liikmete ja töötajate tõhusat haldamist ja omavahelist suhtlemist.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 27 leheküljel, 7 peatükki, 6 joonist ja 0 tabelit.

# Abstract

## Development and Implementation of an Information System for Unions

The aim of this bachelor's thesis is to develop an information system for unions, using the Estonian Railway Workers' Union (ERAÜ) as an example. The information system is divided into three main components: homepage, employee web application, and members' Android application. The homepage provides an overview of the union, allowing users to read news, ask questions, and join the union. The employee web application allows you to manage news, members and employees, and generate reports. The Android application allows members to view their membership cards electronically and enables viewing and editing of personal data and contact information.

In this thesis, front-end applications are developed using the ReactJS framework, JavaScript, and Bootstrap. A PostgreSQL database is created for the homepage, which contains news, answers, and member requests. A separate PostgreSQL database is created for the data of the employee web application, and existing data is migrated to the new database. The back-end application uses Spring Boot and Java version 17. The members' Android application is developed using Android Studio and Java version 8.

As a result of this work, an innovative homepage, a web application for employees, and an Android application will be completed, allowing effective management and communication between ERAÜ members and employees.

The thesis is written in Estonian and contains 27 pages of text, 7 chapters, 6 figures, and 0 tables.

## Lühendite ja mõistete sõnastik

ACID-compliant	Atomicity, Consistency, Isolation, Durability, andmebaasisüsteemi omadused
API	API Application Programming Interface, rakendusliides
CI/CD	Continuous Integration and Continuous Development, pidev integratsioon ja pidev juurutamine
CSS	Cascading Style Sheets, astmelised stiililehed
CSV	Comma Separated Values, failitüüp
DOM	Document Object Model, dokumendi objektimudel
ERAÜ	Eesti Raudteelaste Ametiühing MTÜ
HTTP	Hypertext Transfer Protocol, andmevahetusprotokoll
IDE	Integrated Development Environment, integreeritud arenduskeskkond
JSON	JavaScript Object Notation, andmevahetusvorming
JWT	JSON Web Token, JSON veebimärk
PHP	Hypertext Preprocessor, programmeerimiskeel
UI	User interface, kasutajaliides
VPS	Virtual Private Server, virtuaalne eraserver

# Sisukord

<b>Joonise loetelu</b>	<b>9</b>
<b>1 Sissejuhatus</b>	<b>10</b>
1.1 Kliendi soovid	10
1.2 Ülesande püstitus	10
<b>2 Projekti kirjeldus</b>	<b>12</b>
2.1 Metoodika	12
2.2 Tööjaotus	13
2.3 Praegune ametiühingu infosüsteem	13
2.3.1 ERAÜ praegune koduleht	14
2.3.2 MS Access 2003 andmebaas	14
2.3.3 Mobiilirakendus liikmepileti kuvamiseks	14
2.3.4 Järeldus	14
2.4 Võimalikud alternatiivid ja nende puudused	14
2.4.1 UnionWare	14
2.4.2 eMembership	15
2.4.3 iMIS	15
2.4.4 Union Impact	15
<b>3 Projektidisain</b>	<b>16</b>
3.1 Tarkvara ja tehnoloogiad	16
3.1.1 Tagarakendus	16
3.1.1.1 Java	16
3.1.1.2 Gradle	16
3.1.1.3 Spring Framework ja Spring Boot	16
3.1.1.4 Spring security	16
3.1.1.5 JSON Web Token	17
3.1.1.6 Swagger	17
3.1.1.7 Lombok	17
3.1.1.8 MapStruct	17
3.1.1.9 Hibernate	17
3.1.2 Andmebaas	17
3.1.3 Esirakendus	18
3.1.3.1 React JS	18

3.1.3.2	Bootstrap . . . . .	18
3.1.3.3	Axios . . . . .	18
3.1.4	Android rakendus . . . . .	18
3.1.4.1	Android Studio . . . . .	18
3.1.4.2	Integreeritud tööriistad ja funktsioonid . . . . .	19
3.2	Tarkvara ja tehnoloogia valimise põhjused . . . . .	19
3.2.1	Kodulehe ja töötaja esirakenduse valimise põhjused: . . . . .	19
3.2.1.1	ReactJS . . . . .	19
3.2.1.2	Bootstrap . . . . .	19
3.2.2	Kodulehe ja töötaja tagarakenduse valimise põhjused: . . . . .	19
3.2.2.1	Java 17 . . . . .	19
3.2.2.2	Spring Boot . . . . .	19
3.2.2.3	Mapstruct . . . . .	19
3.2.2.4	Lombok . . . . .	20
3.2.2.5	Spring Security . . . . .	20
3.2.2.6	JWT (JSON Web Token) . . . . .	20
3.2.2.7	Swagger . . . . .	20
3.2.2.8	Gradle . . . . .	20
3.2.2.9	Hibernate . . . . .	20
3.2.3	Kodulehe ja töötaja andmebaasi valimise põhjus . . . . .	20
3.2.4	Android mobiilirakenduse valimise põhjused: . . . . .	20
3.2.4.1	Android Studio . . . . .	20
3.2.4.2	Java 8 . . . . .	21
3.3	Alternatiivsed tehnoloogiad . . . . .	21
3.4	Alternatiivsete tehnoloogiate mitte valimise põhjused . . . . .	21
3.4.1	Kodulehe ja töötaja esirakenduse alternatiivid . . . . .	21
3.4.1.1	Angular . . . . .	21
3.4.1.2	Vue.js . . . . .	21
3.4.1.3	WordPress . . . . .	22
3.4.2	Kodulehe ja töötaja tagarakenduse alternatiivid . . . . .	22
3.4.2.1	Node.js koos Expressiga . . . . .	22
3.4.2.2	Ruby on Rails . . . . .	22
3.4.2.3	Django . . . . .	22
3.4.2.4	ASP.NET Core . . . . .	23
<b>4</b>	<b>Projekti sisu . . . . .</b>	<b>24</b>
4.1	Rollid süsteemis . . . . .	24
4.2	Funktsionaalsed nõuded . . . . .	24
4.3	Mitiefunktsionaalsed nõuded . . . . .	25

4.4	Töö planeerimine ja maht . . . . .	25
4.5	Arendus . . . . .	26
4.5.1	Koduleht . . . . .	26
4.5.1.1	Liitumine . . . . .	26
4.5.1.2	Kontaktvorm . . . . .	26
4.5.1.3	Uudiste ja sündmuste kuvamine . . . . .	26
4.5.1.3.1	Uudiste vaatamine . . . . .	26
4.5.1.3.2	Sündmuste kalender . . . . .	27
4.5.2	Töötaja veebirakendus . . . . .	27
4.5.2.1	Liikmete haldamine . . . . .	27
4.5.2.1.1	Liikmete lisamine ja vastuvõtmine . . . . .	27
4.5.2.1.2	Liikmete teabe muutmine . . . . .	27
4.5.2.1.3	Liikmete kustutamine . . . . .	27
4.5.2.1.4	Aruannete genereerimine . . . . .	28
4.5.2.1.5	Liikmete statistika . . . . .	28
4.5.2.2	Töötajate haldamine . . . . .	28
4.5.2.2.1	Töötajate lisamine . . . . .	28
4.5.2.2.2	Töötajate andmete muutmine . . . . .	28
4.5.2.2.3	Töötajate kustutamine . . . . .	28
4.5.2.3	Liikmemaksu haldamine . . . . .	29
4.5.2.3.1	Liikmemaksu lisamine . . . . .	29
4.5.2.3.2	Liikmemaksu ajaloo vaatamine . . . . .	29
4.5.2.3.3	Maksmata liikmemaksude jälgimine . . . . .	29
4.5.2.4	Uudiste haldussüsteem . . . . .	29
4.5.2.4.1	Uudiste lisamine . . . . .	29
4.5.2.4.2	Uudiste haldamine . . . . .	29
4.5.3	Android rakendus . . . . .	29
4.5.3.1	Sisselogimine . . . . .	29
4.5.3.2	Mitmekeelsus . . . . .	30
4.5.3.3	Liikmekaardi kontroll . . . . .	30
4.5.3.4	Profili haldamine . . . . .	30
4.5.3.5	Parooli muutmine . . . . .	30
4.5.4	Andmebaasi migreerimine . . . . .	30
4.5.4.1	Vana andmebaasi kirjeldus . . . . .	30
4.5.4.2	Migreerimise protsess . . . . .	31
4.6	Ülesande lahendamise voog . . . . .	31
4.7	Rakenduse publitseerimine . . . . .	31
4.7.1	GitLab . . . . .	32
4.7.2	Arenduse protsess . . . . .	32



4.7.3	Gitlab CI/CD . . . . .	32
4.7.4	Docker . . . . .	32
4.7.4.1	Docker Compose . . . . .	32
4.7.5	Nginx ja reverse proxy . . . . .	32
<b>5</b>	<b>Valideerimine . . . . .</b>	<b>34</b>
5.1	Valideerimise protsess meie projektis . . . . .	34
5.2	Testimisviisid ja tagasiside kogumine . . . . .	34
5.3	Järeldus . . . . .	34
<b>6</b>	<b>Tulemused . . . . .</b>	<b>35</b>
<b>7</b>	<b>Edasiarendus . . . . .</b>	<b>37</b>
7.1	Mobiilirakenduste loomine iOSi jaoks . . . . .	37
7.2	Automaatsete e-kirjade saatmine liikmetele . . . . .	37
	<b>Kasutatud kirjandus . . . . .</b>	<b>38</b>
	<b>Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks . . . . .</b>	<b>40</b>
	<b>Lisa 2 – Töötajate rakenduse andmebaasi kontseptuaalne andmemudel . . . . .</b>	<b>41</b>
	<b>Lisa 3 – Autoriseerimise skeem . . . . .</b>	<b>43</b>
	<b>Lisa 4 – Andmete uude andmebaasi importimise kood . . . . .</b>	<b>44</b>

## Jooniste loetelu

1	<i>Töötaja olemi-suhte diagramm.</i> . . . . .	41
2	<i>Liikme olemi-suhte diagramm.</i> . . . . .	42
3	<i>Isiku olemi-suhte diagramm.</i> . . . . .	42
4	<i>Kasutajakonto olemi-suhte diagramm.</i> . . . . .	42
5	<i>Autoriseerimise skeem.</i> . . . . .	43
6	<i>Andmete uude andmebaasi importimise kood.</i> . . . . .	44

# 1. Sissejuhatus

Ametiühingud on organisatsioonid, mis esindavad töötajate huve ja õigusi, pakkudes neile tuge tööga seotud küsimustes. Ametiühingu eesmärk on parandada töötingimusi, tagada töötajatele õiglane töötasu ja kaitsta nende vabadust. Selleks, et ametiühing saaks oma liikmeid tõhusalt esindada ja hallata, vajab ta kaasaegseid tehnoloogiaid ja tarkvaralahendusi.

Käesoleva bakalaureusetöö eesmärk on luua uus koduleht, töötajate veebirakendus ning liikmete Android rakendus, mida saavad kasutada kõik ametiühingud. Selleks kasutatakse nüüdisaegseid tehnoloogiaid, nagu ReactJS, Spring Boot, PostgreSQL ja Java 17. Infosüsteem luuakse lähtuvalt ERAÜ nõuetest ja vajadustest. Koduleht peaks olema uueaegse visuaalse kujundusega ja sisaldama asjakohast sisu, nagu ametiühingu teave, viimased uudised, liikmeks astumise vorm ja kontaktvorm. Töötaja veebirakendus peab olema võimeline haldama uudiseid, kontakte ja liikmemaksu, samuti pakkuma liikmete ja töötajate haldus- ja aruandlusfunktsioone. Liikmete Android rakendus peab võimaldama liikmetel ja töötajatel vaadata ja muuta oma liikmekaarti ning pääseda ligi oma isiku- ja kontaktandmetele. Samamoodi peab see olema suuteline neid andmeid ajakohastama.

Praegune olukord näitab, et ametiühingul on 2011. aastal loodud veebileht, mille kujundus ja teave on vananenud. Töötajate rakendus on MS Access 2003 rakendus, mis töötab vanas arvutis ja haldab liikmeid ja liikmemaksu. Käesoleva bakalaureusetööga tahame luua uusi ja kaasaegseid lahendusi, mis võimaldavad ametiühingul paremini toimida ja pakkuda oma liikmetele kvaliteetset teenust.

## 1.1 Kliendi soovid

Klient soovib täiustatud infosüsteemi, mis võimaldab tõhusat suhtlust liikmete ja töötajate vahel, annab liikmetele täpset ja asjakohast teavet, ning millel on kasutajasõbralik ja ajakohane kasutajaliides. Lisaks soovib ta süsteemi, mis võimaldab tõhusat liikmete haldamist, sealhulgas selliseid funktsioone nagu uudiste uuendamine, küsimuste ja vastuste esitamine ning liikmekaartide elektrooniline kuvamine.

## 1.2 Ülesande püstitus

Käesoleva bakalaureusetöö eesmärk on luua uus koduleht, töötajate veebirakendus ning liikmete Android rakendus, mida saavad kasutada kõik ametiühingud. Selleks kasutatakse

nüüdisaegseid tehnoloogiaid, nagu ReactJS, Spring Boot, PostgreSQL ja Java 17. Infosüsteem luuakse lähtuvalt ERAÜ nõuetest ja vajadustest. Koduleht peaks olema uueaegse visuaalse kujundusega ja sisaldama asjakohast sisu, nagu ametiühingu teave, viimased uudised, liikmeks astumise vorm ja kontaktvorm. Töötaja veebirakendus peab olema võimeline haldama uudiseid, kontakte ja liikmemaksu, samuti pakkuma liikmete ja töötajate haldus- ja aruandlusfunktsioone. Liikmete Android rakendus peab võimaldama liikmetel ja töötajatel vaadata ja muuta oma liikmekaarti ning pääseda ligi oma isiku- ja kontaktandmetele. Samamoodi peab see olema suuteline neid andmeid ajakohastama.

## 2. Projekti kirjeldus

Enne tegelikku arendustegevust loodi projekti üldine raamistik:

- Rollid ja kohustused jagati meeskonnaliikmete vahel
- Määrati kindlaks metoodika arendamise reeglid ja kokkulepped
- Määratleti keskkonnad

Lepiti kokku, et Deniel vastutab tagarakenduse ja andmebaasi arendamise eest. Tarvo vastutab esirakenduse arendamise eest. Karl-Gustav teeb fullstacki. Kuid aja jooksul tuli kliendilt mitmeid uusi taotlusi ning otsustati, et jaotame rollid ja vastutused ümber. Kuna klient soovis telefoni rakendust, pidasime ühise koosoleku ja leidsime lahenduse, kus Deniel vastutab endiselt tagarakenduse ja andmebaasi eest, Tarvo endiselt esirakenduse arendamise eest aga Karl-Gustav tegeleb liikmete Android rakenduse arendamisega.

Projektijuhtimiseks valiti GitLab. GitLab valiti seetõttu, kuna see on ülikooli GitLabiga kõige sarnasem, ning meeskonna liikmed õppisid seda ülikoolis kasutama ning see võimaldab mugavat ja lihtsat koodihaldust. Ülikooli GitLab jäeti välja, sest projekti liikmed kaotavad pärast ülikooli lõpetamist koodi haldamise õigused ja andmed ülikooli Gitlabis.

Discordi kasutati suhtluskanalina nii teksti- kui ka heli/video suhtlemiseks. Peamiseks põhjuseks oli, kuna kõik projektis osalejad olid varasemalt Discordi kasutanud, ning oskasid seda hästi käsitleda. Lisaks sellele võimaldab mõistlikult hästi vormistatud koodi sisestamist/esitamist, ning see on tasuta.

Kohtumised TalTechi juhendajaga oli korraldatud Teamsis, et saada tagasisidet projekti kohta.

Suhtlemine kliendiga toimus e-posti ja telefoni teel.

### 2.1 Metoodika

Tagarakenduste loomiseks kasutatakse Spring Boot-i ja programmeerimiskeelt Java 17. Nii liikme esirakenduse kui ka töötaja esirakenduse loomiseks kasutatakse ReactJS raamistiku, programmeerimiskeelt JavaScript ja teeki Bootstrap. Mõlemaks andmebaasiks kasutatakse

PostgreSQLi versiooni 15. Androidi rakenduse loomiseks kasutatakse Android Studiot. Nii tagarakendusse kui ka esirakendusse võivad lisanduda teegid vastavalt vajadustele.

Arenduse käigus kasutatakse koodihalduseks ja versiooni kontrolliks GitLabi.

Koduleht, selle tagarakendus ja andmebaas pannakse tööle veebiserveris VPSis. Töötaja esirakendus, tagarakendus ja isikute andmebaas pannakse tööle teises veebiserveris VPSis.

## **2.2 Tööjaotus**

Bakalaureusetöö projektis on kolm peamist arendusvaldkonda: backend, frontend ja Android rakendus. Tööjaotus on järgmine:

1. Deniel Konstantinov: Deniel vastutab peamiselt backend arenduse eest, sealhulgas tagarakenduse loomise eest nii liikme veebirakendusele kui ka töötaja veebirakenduse loomise eest. Samamoodi tegeleb Deniel andmebaasi disaini ja arendusega ning olemasolevate andmete migreerimine. Deniel abistab ka Androidi rakenduse arendamisel.
2. Karl-Gustav Rüütli: Karl-Gustav vastutab peamiselt Android rakenduse eest. Ta tegeleb rakenduse kujundamise, arendamise ja testimisega. Lisaks abistab ta Tarvot vajadusel ka frontendi arendamisel.
3. Tarvo Tali: Tarvo keskendub peamiselt frontend arendusele, sealhulgas kodulehe ja töötajate veebirakenduse kujundamisele ja arendamisele. Lisaks töötab koos Denieliga backend integratsiooni ja kasutajaliidese loomisel.

Kogu projekti vältel suhtlevad projekti liikmed omavahel ja ERAÜ esindajatega, et tagada süsteemi nõuete täitmine ja hoida end kursis projekti edenemisega. Lõputöö edenedes võib tööplaanis toimuda muudatusi, mis sõltuvad ERAÜ soovidest ja lõputööga seotud probleemide lahendamisest. Kogu töö käigus on oluline tagasiside ja suhtlemine lõputöö liikmete vahel ja ERAÜ esindajatega, et tagada süsteemi vastavus nõuetele ja ootustele.

## **2.3 Praegune ametiühingu infosüsteem**

Enne projekti alustamist uuris lõputöö meeskond olemasolevaid lahendusi, et saada põhjalikum ülevaade ERAÜ nõuetest ja tuvastada võimalikud probleemid, mis vajavad lahendamist. Need uuringud aitasid välja töötada selgema ettekujutuse sellest, millist tüüpi süsteem tuleks ERAÜ jaoks välja töötada.

### **2.3.1 ERAÜ praegune koduleht**

ERAÜ-l on juba olemas veebileht, mis annab teavet liidu kohta ja võimaldab liikmetel tutvuda selle tegevusega. Praegune veebileht on siiski vananenud ja sisaldab vananenud teavet. Uue veebilehe eesmärk on pakkuda moodsamat kujundust, paremat kasutatavust ja lisafunktsioone ametiühingu tegevuse toetamiseks.

### **2.3.2 MS Access 2003 andmebaas**

ERAÜ kasutab praegu isikuandmete haldamiseks MS Access 2003 andmebaasi. See süsteem on vananenud ja ei paku kõiki vajalikke funktsioone, nagu aruandlust ja integreerimist teiste süsteemidega. Lõputöö raames viiakse andmed üle uude PostgreSQL andmebaasi, mis on kaasaegsem ja pakub rohkem võimalusi andmete haldamiseks ja töötlemiseks.

### **2.3.3 Mobiilirakendus liikmepileti kuvamiseks**

Praegu puudus ERAÜ-l liikmekaardi elektrooniline kuvamine.

### **2.3.4 Järeldus**

Uurides olemasolevaid lahendusi, sai parema ülevaate ERAÜ vajadustest ning võimalikest lahendustest, mida uus süsteem peaks pakkuma. See aitas luua plaani, kuidas arendada süsteemi, mis vastab ERAÜ ootustele ja vajadustele.

## **2.4 Võimalikud alternatiivid ja nende puudused**

Selles peatükis vaadeldakse erinevaid turul saadaolevaid ametiühingu tarkvaralahendusi. Iga alternatiivi puhul tuuakse välja nende peamised võimalused ja funktsioonid ning analüüsitakse, miks need ei vasta ERAÜ vajadustele.

### **2.4.1 UnionWare**

UnionWare on spetsiaalselt ametiühingute jaoks loodud tarkvaralahendus, mis võimaldab hallata liikmete andmeid, ajakavasid, rahaasju, üritusi, võrgustike loomist [1]. Tegemist on modulariseeritud süsteemiga, mis võimaldab ametiühingutel valida lahendusi vastavalt oma vajadustele. Kuigi UnionWare on terviklik lahendus, ei vasta see täielikult ERAÜ nõuetele. UnionWare on tasuline tarkvara, mis muudab selle kasutamise ERAÜ jaoks

kalliks, seetõttu UnionWare ei olnud sobilik valik.

### **2.4.2 eMembership**

eMembership on WinMill Software'i pilvepõhine lahendus, mis võimaldab ametiühingutel hallata liikmete arvu ja suurendada nende kaasamist [2]. See sisaldab erinevaid funktsioone, nagu liikmete registreerimine, maksude haldamine, õiguste haldamine ja palju muud. Kuigi eMembership on alati kättesaadav, on sellel ka puudused. Esiteks on eMembership sarnaselt UnionWare'ile tasuline lahendus, mis suurendab ametiühingu tegevuskulusid. Teiseks ei ole see avatud lähtekoodiga, mis tähendab, et seda ei saa kohandada vastavalt ERAÜ nõuetele.

### **2.4.3 iMIS**

iMIS on mitmeotstarbeline integreeritud infosüsteem, mis pakub organisatsioonidele, sealhulgas ametiühingutele, mitmeid võimalusi [3]. Nende hulka kuuluvad liikmete haldamine, ürituste haldamine, kodulehe loomine ja liikmete kaasamine. iMIS võimaldab luua personaalse liikmekogemuse, suurendades liikmete rahulolu ja kaasamist. Siiski on iMISil ka mõned piirangud. Nagu UnionWare ja eMembership, on ka iMIS tasuline ja mitte avatud lähtekoodiga lahendus, mis piirab selle kohandatavust ERAÜ vajadustele.

### **2.4.4 Union Impact**

Union Impact on tarkvaraplatvorm, mis pakub mitmeid lahendusi ametiühingutele, nagu liikmete haldamine, veebisaidi haldamine, ajakava haldamine, finantside haldamine [4]. See on loodud olema intuiitvne ja kasutajasõbralik. Kuid nagu eespool kirjeldatud lahendustele, ei vasta ka Union Impact ERAÜ nõuetele. Selle põhjuseks on suletud lähtekood.



## **3. Projektidisain**

### **3.1 Tarkvara ja tehnoloogiad**

#### **3.1.1 Tagarakendus**

Infosüsteemi tagarakendused on üles ehitatud kasutades programmeerimiskeelt Java versioon 17 ja Spring Boot-i raamistikku. Infosüsteemil on kaks tagarakendust, millest üks vastutab kodulehe funktsionaalsuse eest ja teine töötajate veebirakenduse funktsionaalsuse eest. Järgnevates alapeatükkides tutvustatakse peamisi tehnoloogiaid, mida kasutatakse infosüsteemi tagarakendustes.

##### **3.1.1.1 Java**

Java on objektorienteeritud programmeerimiskeel, mis sobib serverirakenduste arendamiseks [5]. Java on platvormist sõltumatu, mis tähendab, et Java rakendused võivad töötada mis tahes operatsioonisüsteemis.

##### **3.1.1.2 Gradle**

Gradle on avatud lähtekoodiga ehitamise automatiseerimise tööriist, mis võimaldab lihtsustada ja automatiseerida Java rakenduste loomist ja pakendamist [6]. Gradle aitab hoida projekti struktureerituna ja tagab, et kõik sõltuvused on korrektselt hallatud.

##### **3.1.1.3 Spring Framework ja Spring Boot**

Spring Framework on avatud lähtekoodiga Java põhine raamistik, mis on mõeldud ettevõtete rakenduste arendamiseks [7]. Spring Boot on Spring-i raamistiku projekt, mis võimaldab lihtsalt ja kiiresti luua iseseisvaid rakendusi [8]. Spring Boot-i kasutamine aitab vähendada mitte vajalikku koodi hulka ja lihtsustab konfiguratsiooni.

##### **3.1.1.4 Spring security**

Spring Security on suure jõudlusega ja kohandatav turvalahendus Java põhiste rakendustele [9]. See pakub autentimist, autoriseerimist ja muud turvafunktsioone. Spring Security-ga saab kaitsta serverirakenduse ressursse ja tagada, et ligipääsu saavad ainult volitatud kasutajad.

### **3.1.1.5 JSON Web Token**

JSON Web Token (JWT) on kompaktne ja enesekontrollitav meetod osapoolte vahelise teabe turvaliseks edastamiseks [10]. JWT-sid kasutatakse autentimise ja autoriseerimise eesmärgil, võimaldades serveril tuvastada kasutaja identiteeti ja kontrollida juurdepääsuõigusi.

### **3.1.1.6 Swagger**

Swagger on avatud lähtekoodiga raamistik, mis on loodud API-de (Application Programming Interfaces) dokumenteerimiseks ja disainimiseks [11]. Seda kasutatakse laialdaselt nii API-de arendamisel kui ka tarbimisel.

### **3.1.1.7 Lombok**

Lombok on Java koodi tööriist, mis pakub hulgaliselt annotatsioone, et vähendada mitte vajalikku (ing. k boilerplate) koodi hulka, mis tavaliselt kaasneb Java klassidega [12].

### **3.1.1.8 MapStruct**

MapStruct on koodigeneraator, mis pakub lihtsat ja tõhusat viisi Java beanide vahelise kaardistamise teostamiseks [13].

### **3.1.1.9 Hibernate**

Hibernate on objekt-relatsioonilise kaardistamise (ORM) raamistik Java keeles. See annab võimaluse tegeleda andmebaasidega kasutades objektorienteeritud paradigmat, sujuvalt teisendades objekte ja nende seoseid andmebaasi tabeliteks. Hibernate võimaldab teha keerukaid päringuid, optimeerida jõudlust ja hallata andmebaasi transaktsioone [14].

## **3.1.2 Andmebaas**

Infosüsteem kasutab andmebaasidena PostgreSQL 15 objekt-relatsioonilist andmebaasisüsteemi [15]. Infosüsteemil on kaks andmebaasi, millest üks vastutab kodulehe info säilitamise eest, näiteks uudised, liikmeavaldused ja küsimused. Teine andmebaas vastutab töötajate veebirakenduse eest, hoides näiteks töötajate andmeid, liikmete andmeid ja liikmemaksu andmeid.

### **3.1.3 Esirakendus**

Infosüsteemi esirakendused on loodud kasutades moodsat JavaScripti raamistikku React 18. Rakenduste disain on teostatud Bootstrap 5 teegi abil.

#### **3.1.3.1 React JS**

React JS on avatud lähtekoodiga JavaScripti raamistik, mis on loodud Meta poolt ja mida kasutatakse kasutajaliideste loomiseks [16]. ReactJS võimaldab luua modulaarseid ja skaleeritavaid veebirakendusi ning pakub komponentide põhise lähenemist ja tõhusat uuenduste haldamist.

#### **3.1.3.2 Bootstrap**

Bootstrap on populaarne avatud lähtekoodiga raamistik, mis aitab arendajatel kiiresti ja lihtsalt luua reageeriva (ing. k responsive) ja mobiilisõbralikke veebisaite [17]. Bootstrap pakub mitmeid eelehitatud komponente, mis muudavad kasutajaliidese disaini ja kujunduse loomise kiireks ja lihtsaks.

#### **3.1.3.3 Axios**

Axios on Promise-põhine HTTP klient andmete asünkroonseks laadimiseks serverist. See on üks enimkasutatud teek andmeside haldamiseks Reacti rakendustes.

### **3.1.4 Android rakendus**

Androidi rakendus pakub funktsioone ja tagab liikmetele sujuva ja mugava kasutajakogemuse.

#### **3.1.4.1 Android Studio**

Android Studio on integreeritud arenduskeskkond (IDE), mida kasutatakse Android-i rakenduste loomiseks. See on Google'i ametlik tööriist ja pakub arendajatele laia valikut funktsioone ja võimalusi, mis lihtsustavad ja kiirendavad rakenduste loomise protsessi. Android Studio toetab Android-i platvormi, pakkudes juurdepääsu Android-i API-dele, teekidele, platvormispetsiifilistele juhistele ja muudele ressurssidele. See võimaldab arendajatel luua kvaliteetseid ja optimeeritud rakendusi, mis töötavad sujuvalt erinevates Android seadmetes.[18]

### **3.1.4.2 Integreeritud tööriistad ja funktsioonid**

Android Studio pakub laia valikut integreeritud tööriistu ja funktsioone, mis aitavad arendajatel koodi redigeerida, testida, vigu tuvastada ja projekti hallata. Lisaks saab Android Studio kohandada, lisades erinevaid pistikprogramme ja laiendusi, mis muudavad arendusprotsessi veelgi lihtsamaks ja kiiremaks.[18]

## **3.2 Tarkvara ja tehnoloogia valimise põhjused**

### **3.2.1 Kodulehe ja töötaja esirakenduse valimise põhjused:**

#### **3.2.1.1 ReactJS**

Valik langes ReactJS-le, kuna see pakub komponendipõhist arhitektuuri ja tõhusaid uuendusi, mis tagavad kiire ja reageeriva kasutajaliidese loomise. Paindlikkus, võimsus ja laiaulatuslik kogukonna toetus olid otsustavad tegurid ReactJS-i eelistamisel Angulari ja Vue.js-i ees.

#### **3.2.1.2 Bootstrap**

Bootstrap valiti kiire ja järjepideva kasutajaliidese arendamise võimaldamiseks. See CSS-raamistik võimaldab luua mobiilisõbralikke ja reageerivaid veebilehti. Selle ühilduvus ReactJS-iga ja lihtne kasutusmugavus olid otsustavad valikutegurid.

### **3.2.2 Kodulehe ja töötaja tagarakenduse valimise põhjused:**

#### **3.2.2.1 Java 17**

Java 17 valiti, kuna see on võimas objektorienteeritud programmeerimiskeel, mis pakub jõudlust, turvauuendusi ja rikkalikke funktsioone. Meie kolmeaastane kogemus Java keelega, stabiilsus ja suur kogukonna toetus olid otsustavad valikutegurid.

#### **3.2.2.2 Spring Boot**

Spring Boot on populaarne raamistik Java-põhiste veebirakenduste arendamiseks, valiti selle ulatusliku dokumentatsiooni ja aktiivse kogukonna toetuse tõttu.

#### **3.2.2.3 Mapstruct**

Mapstruct valiti, et lihtsustada erinevate Java bean tüüpide vahelist konverteerimist, suurendades arenduse produktiivsust ja hooldatavust.

#### **3.2.2.4 Lombok**

Lombok valiti, et vähendada mitte vajalikku koodi, automaatkoodi genereerimise võimaldamiseks, parandades koodi loetavust ja arendamise lihtsust.

#### **3.2.2.5 Spring Security**

Spring Security valiti, et pakkuda rakendusele vajalikke turvalisusfunktsioone.

#### **3.2.2.6 JWT (JSON Web Token)**

JWT valiti, kuna seda on lihtne kasutada ja implementeerida, ning lisaks oli meeskonnal eelnev hea kogemus sellega.

#### **3.2.2.7 Swagger**

Swagger valiti, kuna see pakub lihtsat ja intuitiivset kasutajaliidest, mis aitab kiiresti genereerida ja visualiseerida API dokumentatsiooni. See aitab ka vähendada arendusaega, kuna see võimaldab testida ja debugida API-sid otse brauseris.

#### **3.2.2.8 Gradle**

Gradle valiti, sest see on paindlik, toetab mitmeid programmeerimiskeeli ja raamistikke, toetab sõltuvuste haldamist ja pakub paremat automatiseerimist.

#### **3.2.2.9 Hibernate**

Hibernate valiti, sest see pakub suurt paindlikkust ja tõhusust andmebaaside töötlemisel ning toetab andmebaasisüsteemide automaatseid ja poolautomaatseid meetodeid.

### **3.2.3 Kodulehe ja töötaja andmebaasi valimise põhjus**

PostgreSQL 15 valiti selle stabiilsuse, jõudluse ja laiendatavuse tõttu, mis muudab selle sobivaks meie lõputöö jaoks.

### **3.2.4 Android mobiilirakenduse valimise põhjused:**

#### **3.2.4.1 Android Studio**

Android Studio valiti kui ametlik IDE Android rakenduste arendamiseks.

### 3.2.4.2 Java 8

Java 8 valiti Androidi arendamiseks, kuna see on laialt levinud ja ühilduv paljude Androidi seadmete ja librarytega.

## 3.3 Alternatiivsed tehnoloogiad

Lisaks valitud tehnoloogiatele kaaluti ka teisi alternatiive järgmiselt:

**Kodulehe ja töötaja esirakenduse loomiseks:** Angular, Vue.js, WordPress

**Kodulehe ja töötaja tagarakenduse loomiseks:** Node.js, Ruby on Rails, Django, ASP.NET Core

**Android mobiilirakenduse loomiseks:** React Native

## 3.4 Alternatiivsete tehnoloogiate mitte valimise põhjused

Alternatiivsete tehnoloogiate mitte valimine toimus põhinedes mitmete kaalutlustele.

### 3.4.1 Kodulehe ja töötaja esirakenduse alternatiivid

#### 3.4.1.1 Angular

**Kogukonna toetus:** Kuigi Angularil on tugev kogukond, eelistati ReactJS-i, millel on suurem kogukond [19].

**Integratsioon backend tehnoloogiatega:** Valitud tagarakenduse tehnoloogiad, nagu Java 17 ja Spring Boot, integreeruvad sujuvalt ReactJS-iga. Angular võis nõuda täiendavat konfigureerimist ja integreerimist, mis tõi esile ReactJS-i ja Spring Boot-i kombinatsiooni eelistuse.

#### 3.4.1.2 Vue.js

**Kogukonna tugi:** Kuigi Vue.js-l kasvav kogukond, on ReactJS-il suurem ja väljakujunenud kogukond, mis pakub laiemat valikut teeke, tööriistu ja ressursse, lihtsustades seeläbi probleemide lahendamist.

**Populaarsus ja kasutuselevõtt:** ReactJS-i populaarsus soodustab selle pikaajalist püsivust ja tagab suure hulga kvalifitseeritud arendajate olemasolu. Kuna oli oluline pikaajaline

hooldus ja teabe kättesaadavus, ei peetud Vue.js-i sobivaks valikuks [19].

### 3.4.1.3 WordPress

**Skaleeritavus ja jõudlus:** WordPress on peamiselt mõeldud sisu ja blogide haldamiseks ning pole parim valik suure jõudlusnõudlusega rakendustele. Lisaks põhineb WordPress PHP programmeerimiskeelel, millel on jõudluse piirangud võrreldes ReactJS-i ja Angulari raamistikega.

**Turvalisuse kaalutlused:** WordPress vajab pidevaid uuendusi, turva pluginaid ja teemade hoolikat haldamist. Kogemuste ja teadmiste puudumine meeskonnas suurendab potentsiaalset riski, eriti kui tegemist on tundlike andmete käitlemisega [20].

**Laiendatavus ja integratsioon:** Kuigi WordPress pakub laia valikut pluginaid ja integratsiooni võimalusi, ei paku see laiendatavuse ja integratsiooni taset, mida ReactJS ja Bootstrap üksteisele pakuvad.

## 3.4.2 Kodulehe ja töötaja tagarakenduse alternatiivid

### 3.4.2.1 Node.js koos Expressiga

Node.js on populaarne JavaScripti jooksutuskeskkond, mis võimaldab serveripoolset arendustegevust [21]. Kombineerituna Expressiga, mis on minimalistlik veebirakenduste raamistik, pakub see kerget ja kiiret lahendust APIdega ja serveripoolsete rakenduste loomisega [22]. Kuid eelistati Javat ja Spring Boot-i, mis on rohkem dokumenteeritud ja millel on suurem kogukond.

### 3.4.2.2 Ruby on Rails

Ruby on Rails on raamistik, mis on tuntud oma lihtsuse ja "konventsioon üle konfigureerimise" lähenemise poolest [23]. Kuid meeskonna puudulike Ruby on Railsi oskuste tõttu eelistati Java põhiseid tehnoloogiaid.

### 3.4.2.3 Django

Django on kõrgetasemeline Pythoni veebiraamistik, mis edendab puhtaid ja tõhusaid arendustavasid [24]. Django ei valitud siiski selle projekti tagarakenduse arenduseks, kuna meeskonnal oli rohkem kogemusi Java ja Spring Boot-iga.

#### **3.4.2.4 ASP.NET Core**

ASP.NET Core on platvormidevaheline raamistik suure jõudlusega veebirakenduste arendamiseks, kasutades C# või F# programmeerimiskeeli [25]. Kuigi ASP.NET Core oli lõputöö jaoks sobiv valik jõudluse poolest, ei valitud seda, kuna meeskonnas puudusid C# ja F# keele oskused ning polnud kindlust, kui hästi see teisi tehnoloogiaid täiendab.



## 4. Projekti sisu

### 4.1 Rollid süsteemis

Süsteemis on määratletud erinevad rollid, mis määravad kasutajate õigused ja kohustused:

1. ERAÜ liige: Ametiühingu liige, kellel on juurdepääs Androidi rakendusele, et vaadata liikmepiletit ning vaadata ja muuta oma andmeid.
2. Töötaja: ERAÜ töötaja, kellel on juurdepääs töötajate veebirakendusele, et hallata liikmeid, uudiseid, küsimusi ning liikmeks astumise avaldusi.
3. Administraator: ERAÜ administraator, kellel on laiendatud õigused töötaja veebirakendusele, sealhulgas uute töötajate lisamine ja olemasolevate töötajate eemaldamine, süsteemi seadistamine ja muude administraatori tasandi ülesannete täitmine.
4. Kodulehe külastaja: ERAÜ kodulehe külastaja, kes saab tutvuda ametiühingu tegevusega, lugeda uudiseid, esitada küsimusi ja täita liikmeks astumise vormi.

### 4.2 Funktsionaalsed nõuded

Süsteemi funktsionaalsed nõuded hõlmavad järgmist:

Kodulehekül:

- Ametiühingu Liidu tegevuse ja ürituste esitlemine
- Uudiste ja teadete kuvamine
- Küsimuste esitamise vorm
- Liikmeks astumise vorm

Töötajate veebirakendus:

- Uudiste lisamine ja haldamine
- Küsimuste ja vastuste haldamine
- Liikmeks astumise avalduste haldamine
- Liikmete ja töötajate haldamine
- Aruannete ja graafikute genereerimine
- Andmete importimine CSV ja EXCEL failidest

Android-rakendus:

- Liikmepileti elektrooniline kuvamine
- Liikmeandmete kontrollimine ja värskendamine
- Liikme parooli vahetamine

### 4.3 Mittefunktsionaalsed nõuded

- Kasutusmugavus: Süsteem peab olema lihtne ja mugav kasutada nii liikmetele, töötajatele kui ka kodulehe külastajatele.
- Skaleeritavus: Süsteem peab olema võimeline toetama tulevikus kasvavat kasutajate arvu ja andmemahtu.
- Turvalisus: Süsteem peab tagama andmete turvalisuse ja privaatsuse ning vastama isikuandmete kaitse nõuetele.
- Jõudlus: Süsteem peab tagama kiiret ja sujuvat kasutajakogemust, et kasutajad saaksid oma ülesandeid kiiresti ja tõhusalt täita.
- Ühilduvus: Süsteem peab ühilduma erinevate seadmete, brauserite ja operatsioonisüsteemidega, et tagada laiaulatuslik kasutatavus.
- Hooldatavus: Süsteemi peab olema lihtne hooldada, uuendada ja vajadusel laiendada.

### 4.4 Töö planeerimine ja maht

Töö planeerimise ja mahu määramiseks on oluline arvestada projekti eesmärke, rollide jaotust, funktsionaalseid ja mittefunktsionaalseid nõudeid ning ressursse. Töö teostamisel tuleb arvestada järgmiste nõuetega:

- Disain ja arhitektuur: Luua süsteemi üldine arhitektuuriline plaan, mis hõlmab kasutatavaid tehnoloogiaid, andmebaasi struktuuri, kasutajaliidese disaini ja suhtlust erinevate komponentide vahel.
- Tööjaotus: Tööülesannete jaotamine meeskonnaliikmete vahel, tagades, et iga liige vastutab konkreetse valdkonna eest ja vajadusel abistab teisi.
- Töö teostamine: Töö teostamine vastavalt plaanile, sealhulgas koodi kirjutamine, testimine, kasutajaliidese loomine ja süsteemi integreerimine.
- Testimine ja kvaliteedi tagamine: Süsteemi põhjalik testimine, et tagada kõigi nõuete täitmine, kasutusmugavus ja jõudlus.
- Dokumentatsioon ja kasutajatugi: Süsteemi tehnilise dokumentatsiooni loomine ning vajadusel kasutajatoe pakkumine.
- Süsteemi kasutuselevõtmine ja hooldus: Süsteemi kasutuselevõtmine, järelevalve ja

korrapärane hooldus, tagab süsteemi nõuetekohase toimimise ja pideva arengu.

## **4.5 Arendus**

### **4.5.1 Koduleht**

#### **4.5.1.1 Liitumine**

Veebisaidil on registreerimise leht, kus üksikisikud saavad liituda ametiühinguga. Küllastajatele esitatakse liikmeks astumise vorm, kus nad peavad sisestama teatud teabe ettenähtud väljadele. Registreerumise turvalisuse ja autentsuse tagamiseks viiakse läbi ka Recaptcha test, et kontrollida, kas kasutaja on inimene. Kui sisselogimisvorm ja Recaptcha test on edukalt läbitud, edastatakse esitatud andmed andmebaasi.

#### **4.5.1.2 Kontaktvorm**

Küllastajate ja ametiühingu töötajate vahelise suhtluse lihtsustamiseks on veebilehel mugav kontaktvorm. See lehekülg on platvormiks, mille kaudu küllastajad saavad võtta otse ühendust ametiühingu töötajatega. Kontaktvormi lehel saavad küllastajad juurdepääsu ametiühingu töötajate kontaktandmetele, et võtta ühendust töötajaga.

Kontaktvormi lehele sisenedes palutakse küllastajatel esitada olulised andmed, sealhulgas oma e-posti aadress, nimi ja sõnumi sisu. See teave tagab, et töötajad saavad tõhusalt vastata nende päringutele ja pakkuda asjakohast abi. Kui küllastaja on sõnumi saatnud, edastatakse see kohe töötajate rakendusse, kus see muutub kättesaadavaks ametiühingu töötajatele.

Ametiühingu töötajad saavad kontrollida regulaarselt töötajate rakendust, et vaadata läbi küllastajatel saadud sõnumeid ja vastata neile. Suhtluse säilitamiseks on töötajatel võimalik vastata küllastaja sõnumile e-posti teel. Kasutades e-posti kui suhtluskanalit, saavad ametiühingu töötajad anda üksikasjalikke vastuseid ja käsitleda küllastaja võimalikke lisaküsimusi või ebaselgusi. Vastus saadetakse seejärel otse küsija e-posti aadressile, mis tagab tõrgeteta ja kiire teabevahetuse.

#### **4.5.1.3 Uudiste ja sündmuste kuvamine**

##### **4.5.1.3.1 Uudiste vaatamine**

Ametiühingu kodulehel on uudiste sektsioon, kus kõik liikmed ja küllastajad saavad lugeda viimaseid uudiseid ja artikleid. Kasutajad saavad uudiseid otsida ja filtreerida erinevate kriteeriumide alusel, et leida neile kõige olulisemat teavet.

#### **4.5.1.3.2 Sündmuste kalender**

Kodulehel on ka interaktiivne kalender, mis kuvab tulevased ja mineviku üritused. Kalender on organiseeritud nii, et täna toimuvad sündmused on eristatud, aidates kasutajatel lihtsasti navigeerida ja leida neile kõige olulisemaid üritusi.

### **4.5.2 Töötaja veebirakendus**

#### **4.5.2.1 Liikmete haldamine**

Infosüsteemi töötaja rakendus võimaldab ametiühingu töötajatel hallata liikmete andmebaasi. Töötaja rakenduse kaudu saab lisada uusi liikmeid, muuta olemasolevate liikmete teavet või kustutada neid süsteemist.

##### **4.5.2.1.1 Liikmete lisamine ja vastuvõtmine**

Süsteem toetab uute liikmete lihtsat lisamist ja vastuvõtmist. Uue liikme lisamise protsess on automatiseeritud, kus on vajalik sisestada vaid olulised andmed. Peale liikme põhiandmete sisestamist genereerib süsteem automaatselt täiendava vajaliku informatsiooni, nagu liikmelisuse alguskuupäev ja liikme seisund.

Süsteem loeb andmeid liikmeavalduse tabelist, mis võimaldab hiirevajutusega salvestada avalduse esitanud isiku andmed andmebaasi. Enne salvestamist on võimalik sisestatud andmeid kontrollida ja vajadusel muuta. See tagab, et andmed on enne salvestamist täpsed ja usaldusväärsed.

Samuti on võimalik lisada uusi liikmeid, täites lihtsalt vormi vastavate andmetega.

Lisaks toetab süsteem liikmete määramist erinevatesse ametiühingu osakondadesse, võimaldades nii paremat liikmete haldust ja organisatsiooni ülesehitust.

##### **4.5.2.1.2 Liikmete teabe muutmine**

Infosüsteem võimaldab lihtsat ja kiiret liikme andmete muutmist. Selle kaudu saab muuta liikme kontaktandmeid ja staatust, lisada või muuta liikme ametiühingu osakondadesse kuuluvust ja teisi asjakohaseid üksikasju.

##### **4.5.2.1.3 Liikmete kustutamine**

Liikmeid saab süsteemist kustutada lihtsa ja selge protsessi kaudu. Enne lõpliku kustu-

tamise kinnitamist küsib süsteem kinnitust, et vältida kogemata kustutamist.

#### **4.5.2.1.4 Aruannete genereerimine**

Süsteem võimaldab luua automaatseid aruandeid liikmete kohta, mis aitavad analüüsida liikmeskonna struktuuri ja arengut. Need aruanded hõlmavad teavet, nagu liikmete arv, liikmete jaotus ettevõtete, osakondade ja elukutsete järgi.

#### **4.5.2.1.5 Liikmete statistika**

Lisaks aruannete genereerimisele võimaldab süsteem reaajas statistikat liikmeskonna kohta. See hõlmab graafikuid ja diagramme, mis kujutavad liikmete jaotust erinevate kategooriate järgi, nagu ettevõtted, elukutsed, ja liikmeskonna vanus.

### **4.5.2.2 Töötajate haldamine**

Infosüsteemi töötaja rakendus võimaldab ametiühingu volitatud isikutel hallata töötajate andmebaasi. Töötaja rakenduse kaudu saab lisada uusi töötajaid, muuta olemasolevate töötajate teavet või kustutada neid süsteemist.

#### **4.5.2.2.1 Töötajate lisamine**

Töötajate lisamine süsteemi on lihtne protsess, mille saab läbi viia ametiühingu juht või volitatud isik, täites selleks ettenähtud vormi. Vorm nõuab töötaja põhiandmete sisestamist, sealhulgas nime, kontaktandmeid, ametinimetust ja töötamise alguskuupäeva.

#### **4.5.2.2.2 Töötajate andmete muutmise**

Süsteem võimaldab paindlikult muuta töötajate andmeid vastavalt vajadusele. See hõlmab nii põhi- kui ka lisainformatsiooni, näiteks kontaktandmete värskendamist või ametinimetuse muutmist. Töötajate andmete muutmise õigus on piiratud teatud rollidega, et tagada andmete turvalisus ja privaatsus.

#### **4.5.2.2.3 Töötajate kustutamine**

Kui töötaja lahkub ametiühingust, on süsteemis võimalik tema andmed eemaldada. Töötajate kustutamise õigus on samuti piiratud, et tagada ainult volitatud isikute juurdepääs sellisele toimingule.

### **4.5.2.3 Liikmemaksu haldamine**

#### **4.5.2.3.1 Liikmemaksu lisamine**

Süsteem pakub kahte viisi liikmemaksu andmete sisestamiseks: manuaalselt või Exceli faili importimise kaudu. Manuaalse sisestamise korral on vajalik sisestada nii liikme andmed kui ka makse summa ja kuupäev. Exceli faili importimisel saab korraga sisestada suure hulga andmeid, vähendades vajadust andmete üksikasjalikuks sisestamiseks.

#### **4.5.2.3.2 Liikmemaksu ajaloo vaatamine**

Süsteem võimaldab vaadata liikmemaksu laekumiste ajalugu, filtreerides andmeid kas ajaperioodi või isiku järgi. See annab kiire ülevaate liikme maksete staatusest ja võimaldab lihtsasti tuvastada võimalikud makseprobleemid.

#### **4.5.2.3.3 Maksmata liikmemaksude jälgimine**

Süsteem pakub võimalust tuvastada ja jälgida liikmeid, kes ei ole maksnud liikmemaksu määratud ajaperioodil. Selline funktsioon võimaldab ametiühingul kiiresti reageerida ja suhelda liikmetega, kes vajavad maksete osas täiendavat tuge või selgitust.

### **4.5.2.4 Uudiste haldussüsteem**

#### **4.5.2.4.1 Uudiste lisamine**

Uudiseid või artikleid saab lihtsasti lisada läbi intuitiivse graafilise redaktori, mis võimaldab lisada ja kujundada teksti ning lisada pilte. See lihtne ja kasutajasõbralik tööriist võimaldab luua professionaalse välimusega uudiseid.

#### **4.5.2.4.2 Uudiste haldamine**

Infosüsteem võimaldab kõiki postitatud uudiseid hallata ühes kohas. Siit saab näha kõiki avaldatud artikleid, muuta nende nähtavust või neid kustutada.

## **4.5.3 Android rakendus**

### **4.5.3.1 Sisselogimine**

Rakenduse peamine funktsioon on turvaline sisselogimine, mis võimaldab ametiühingu liikmetel pääseda ligi oma kontole, hallata oma andmeid, vaadata isiklikku andmeid ja liikmekaarti. Sisselogimine toimub e-posti aadressi ja salasõna abil, tagades kasutajate andmete turvalisuse.

#### **4.5.3.2 Mitmekeelsus**

Arvestades, et ametiühingu liikmed võivad rääkida erinevaid keeli, on rakenduse mitmekeelsus oluline funktsioon. See võimaldab kasutajatel valida eelistatud keele ja vahetada seda lihtsasti, pakkudes ligipääsu nendele liikmetele, kes räägivad ainult ühte keelt.

#### **4.5.3.3 Liikmekaardi kontroll**

Rakendus võimaldab liikmetel kontrollida oma liikmekaardi kehtivust ja näidata seda vajadusel. See sisaldab teavet liikmekaardi kehtivuse kohta ning võimaldab liikmetel tõendada oma kuulumist ametiühingusse ilma andmebaasi või nimekirja kaudu automaatselt kontrollimata.

#### **4.5.3.4 Profili haldamine**

Kasutajatel on võimalus hallata ja muuta oma profiili, sealhulgas isikliku teabet ja kontaktandmeid. See tagab, et kasutajate andmed on alati ajakohased ja korrektsed.

#### **4.5.3.5 Parooli muutmine**

Rakendus pakub liikmetele võimalust muuta oma konto parooli, sisestades kaks korda uue parooli pärast vana parooli esitamist või esitada oma email ja saadud kood emailist sisestada ja uus parool samuti. See võimaldab kasutajatel oma konto turvalisuse tagamiseks perioodiliselt paroole vahetada.

### **4.5.4 Andmebaasi migreerimine**

#### **4.5.4.1 Vana andmebaasi kirjeldus**

Arendusprotsessi käigus tuli silmitsi seista olulise väljakutsega: olemasoleva andmebaasi struktuuriga. Kliendi poolt kasutatav andmebaas oli loodud MS Access 2003 tarkvaraga, mis on aja jooksul muutunud vanaks. Selle tarkvara poolt toetatud andmebaasi formaat ja funktsioonid ei ole päris kooskõlas tänapäevaste nõuetega, mis tekitas teatud raskusi andmebaasi uuendamisel.

Lisaks formaadi vananemisele oli olemasoleva andmebaasi struktuur probleemne. Üks märkimisväärsemaid probleeme oli see, et alamtabelites oli korduvaid lahtreid, mis rikkusid normaalvormide põhimõtteid ning muutsid andmete tõhusa haldamise keeruliseks. Korduvad andmed põhjustasid andmete dubleerimist.

#### 4.5.4.2 Migreerimise protsess

Andmebaaside migreerimine algas tabelite eksportimisest Exceli failidena. Seejärel loodi Pythoni kood (Lisa 4), mis luges failid sisse ja lõi uue tabeli ajutises skeemis uues andmebaasis koos andmetega. Järgnevalt kirjutati SQL laused, millega tõsteti kõik vajalikud andmed uutesse tabelitesse, kus see oli võimalik. Mõnedes lahtrites esines ebatäpset või korduvat infot, mis salvestati veerus "additionalInformation". Hiljem tõsteti see teave käsitsi sobivatesse tabelitesse. Selline automaatne protsess polnud võimalik, kuna eelnevas andmebaasis puudusid kontrollid mõnede veergude suhtes. Seetõttu sisaldasid andmed palju kirjavigu ja vigaseid andmeid.

## 4.6 Ülesande lahendamise voog

Kasutaja suhtlemine süsteemiga:

- Kasutaja logib süsteemi sisse, kasutades autentimisprotsessi.
- Kasutaja navigeerib rakenduse erinevate funktsioonide vahel, otsides vajalikku teavet või tegevust.
- Kasutaja sisestab andmed või teostab tegevuse (näiteks lisab uue liikme, muudab olemasoleva liikme andmeid, vastab küsimusele ja nii edasi).
- Töötaja tagarakendus suhtleb kodulehe tagarakendusega, et teostada vajalikud päringud ja tegevused.
- Tagarakendused töötlevad andmeid ja suhtlevad andmebaasidega, et salvestada või pärida vajalikku teavet.
- Töötaja tagarakendus saab tulemusel kodulehe tagarakenduselt ja edastab need töötaja esirakendusele.
- Kasutaja saab süsteemist tagasisidet, mis kinnitab tegevuse edukust või ebaõnnestumist, ja näeb tulemusi.

Kogu protsessi käigus tagab süsteem turvalise ja kiire kasutuskogemuse, pakkudes kasutajatele vajalikku teavet ja abistades neid ülesannete täitmisel.

## 4.7 Rakenduse publitseerimine

See rakendus on avatud lähtekoodiga ning selle arendamine ja salvestamine toimub GitLabi platvormil. Kogu arendusprotsess on kaetud, alates koodi kirjutamisest ja testimisest kuni versioonihalduseni.



### **4.7.1 GitLab**

GitLab on versioonikontrolli- ja koostööplatvorm, mis toetab efektiivset projektijuhtimist ja meeskonnatööd[26]. GitLabi kasutatakse koodi salvestamiseks, ülesannete jälgimiseks ja projekti töövoogude lihtsustamiseks.

### **4.7.2 Arenduse protsess**

Arendusprotsess koosneb koodi kirjutamisest, testimisest, integreerimisest ja rakenduse versiooni haldamisest. GitLabi abil saavad meeskonnaliikmed jagada koodimuudatusi, jälgida ülesandeid ja hoida üksteist kursis.

### **4.7.3 Gitlab CI/CD**

GitLab CI/CD (Continuous Integration/Continuous Deployment) on automatiseeritud protsess, mis tagab, et kood töötab korralikult enne selle tootmisse laskmist. See sisaldab koodi testimist, integreerimist ja vabastamist (ing. k release). GitLab CI/CD abil saate automatiseerida koodi testimist, pakkimist ja vabastamist, vähendades käsitsi tehtavaid toiminguid ja kiirendades arendustegevust [27].

### **4.7.4 Docker**

Docker on konteineritehnoloogia, mis tagab rakenduste kiire ja lihtsa täitmise erinevates keskkondades, paigutades rakendused koos nende sõltuvustega. Dockeri kasutamine tagab rakenduse järjepideva jõudluse erinevates keskkondades ja lihtsustab rakenduse käivitamist [28].

#### **4.7.4.1 Docker Compose**

Docker Compose on tööriist, mis võimaldab konfigureerida ja käivitada mitu Dockeri konteinerit ühe faili abil. See muudab konteinerite haldamise lihtsamaks ja kiirendab keskkonna seadistamist [29].

### **4.7.5 Nginx ja reverse proxy**

Nginx on võimas HTTP-server ja reverse proxy, mis edastab võrgupäringud asjakohastele rakendustele või teenustele. Nginxi reverse proxy abil saab kasutajate päringud suunata õigetele komponentidele, mis võimaldab turvalisust, koormuse tasakaalustamist ja paremat

kontrolli. Nginx on tuntud oma stabiilsuse, jõudluse ja paindlikkuse poolest, mistõttu on see sobiv valik selliseks süsteemiks. [30]

## **5. Valideerimine**

Valideerimine on ametiühingu süsteemi arendusprotsessis kriitiline samm, mis mängib olulist rolli süsteemi funktsionaalsuse, terviklikkuse ja kasutajate rahulolu tagamisel. Selles etapis kasutatakse erinevaid testimis- ja tagasisidemeetodeid, et tagada süsteemi kvaliteet, usaldusväärsus ja kasutatavus.

### **5.1 Valideerimise protsess meie projektis**

Antud lõputöö puhul kasutati valideerimisprotsessis erinevaid lähenemisviise. Esiteks tagati disaini kvaliteedi, küsides tagasisidet kasutajaliideste õppejõult. Õppejõud andis mitmeid väärtuslikke soovitusi, sealhulgas ettepanekuid tekstifontide, värviskeemide, kontrasti ja veebilehe elementide paigutuse kohta.

Samuti võeti tagasiside saamiseks ühendust ERAÜ töötajate ja juhatuse liikmetega. Küsimused käsitlesid nii visuaalseid aspekte, nagu näiteks värviskeemid, kui ka sisulist teavet, näiteks mida nad sooviksid veebilehel näha. Eriti oluliseks osutus töötajate tagasiside töötajate rakenduse sisu ja funktsionaalsuse kohta.

### **5.2 Testimisviisid ja tagasiside kogumine**

Kasutajate testimine ja tagasiside kogumine oli valideerimisprotsessi üks osadest. Kasutajate testimine võimaldas hinnata ametiühingu süsteemi kasutatavust, kasutajakogemust ja funktsionaalsust lõppkasutaja seisukohast.

### **5.3 Järeldus**

Valideerimine on ametiühingu süsteemi arendamise üks etappidest, mis tagab süsteemi kvaliteedi, usaldusvääruse ja kasutatavuse. Antud lõputöös näitas see protsess selgelt, et konstruktiivne tagasiside on süsteemi disaini ja funktsionaalsuse parandamiseks hädavajalik. Pärast valideerimisprotsessi hoolikat läbiviimist suudeti luua usaldusväärse, kasutajasõbraliku ja toimiva platvormi, mis vastab ERAÜ liikmete, töötajate ja küllastajate vajadustele.

## 6. Tulemused

Lõputöö raames realiseeriti ametiühingute vajadustele kohandatud ja täiustatud infosüsteem, keskendudes Eesti Raudteetöötajate Ametiühingule. See saavutati kaasaegse kodulehe, töötajatele mõeldud põhjaliku veebirakenduse ning kasutajasõbraliku ja lihtsa liikme Android-rakenduse loomisega. Valminud infosüsteem vastab ametiühingute vajadustele, pakkudes vajalikke vahendeid ja funktsioone, et lihtsustada tegevust ja parandada liikmetele pakutavaid teenuseid.

Uus koduleht on visuaalselt kutsuv ja informatiivne platvorm, mis annab küllastajatele põhjaliku ülevaate ametiühingu tegevusest ja üritustest. See pakub ajakohaseid uudiseid, julgustab küllastajatega suhtlemist ja lihtsustab olulise teabe jagamist. Lisaks sisaldab koduleht küsimuste esitamismuud, kus küllastajad saavad esitada küsimusi ja saada vastuseid ametiühingu töötajatelt. Lihtsustatud liikmemuud teeb ametiühingu liitumise protsessi sujuvamaks.

Töötajate veebirakendus on oluline vahend ametiühingu siseste protsesside ühtlustamiseks. Töötaja rakenduse funktsioonide hulka kuulub uudiste ja teadete efektiivne haldamine, mis tagab, et asjakohane teave jõuab igaheni õigeaegselt. Küsimuste ja vastuste haldamise funktsioonid lihtsustavad kasutajate päringute töötlemist, võimaldavad kiireid vastuseid ja edendavad kaasamist. Töötaja rakendus võimaldab liikmete taotluste sujuvat haldamist, lihtsustab uute liikmete liitumist ja võimaldab mitteaktiivsete kirjade kustutamist. Süsteem võimaldab terviklikku liikmete ja töötajate haldamist, kergendab suhtlust ja koostööd ning tagab täpse arvestuse. Aruannete koostamine ja diagrammide kuvamine annab kasuliku ülevaate ametiühingu tegevusest, aidates kaasa strateegilisele planeerimisele ja otsuste tegemisele. Võimalus importida andmeid CSV- ja Exceli-failidest parandab andmete integreerimist, kõrvaldades käsitsi sisestatavad andmed ja suurendades üldist tulemuslikkust.

Androidi rakendus on liikmetele ja töötajatele mugav vahend, mille abil saavad nad oma mobiilseadmete kaudu juurdepääsu olulisele teabele ja teenustele. Elektrooniline liikmekaart kaotab vajaduse füüsiliste kaartide järele ning pakub tänapäevast ja keskkonnasõbralikku lahendust. Liikmed saavad mugavalt vaadata oma liikmesuse teavet, pääsedes ligi isikuandmetele ja kontaktandmetele. Võimalus vaadata ja uuendada isiklikke andmeid liikvel olles pakub lisamugavust ja iseteenindusvõimalusi.

Teostamise käigus pöörati erilist tähelepanu mitte funktsionaalsetele nõuetele. Infosüsteem on kasutajasõbralik ja pakub kõigile kasutajatele intuitiivset kasutajaliidest. Oluline kriteerium on laiendatavus, et tagada, et süsteem suudab toime tulla kasutajate kasvava nõudluse ja kasvava andmemahuga, ilma et see kahjustaks jõudlust. Andmete kaitsmiseks ja eraelu puutumatuset käsitlevate nõuete täitmiseks on kehtestatud turvameetmed. Süsteem pakub suurepäraselt jõudlust, mis tagab kasutajakogemuse sujuvuse. Ühilduvus erinevate seadmete, brauserite ja operatsioonisüsteemidega tagab laialdase kättesaadavuse ja kasutusmugavuse.

Süsteemi ülesehitus ja disain kergendavad hooldust, uuendusi ja tulevase täiustusi. Läbimõeldud ja hästi struktureeritud süsteemiarhitektuur võimaldab efektiivset süsteemi haldamist ja uute funktsioonide või täienduste lihtsat integreerimist, kui see on vajalik. See aitab kaasa süsteemi jätkusuutlikkusele ja pikaajalisusele ning tagab selle jätkuva asjakohasuse ja väärtuse ametiühingu jaoks.

## **7. Edasiarendus**

### **7.1 Mobiilirakenduste loomine iOSi jaoks**

Järgmine samm oleks luua iOS-rakendus. See oleks oluline edasiarendus, kuna praegune mobiilne lahendus on saadaval ainult Android platvormil. Kuigi Android on laialdaselt kasutusel, kasutab suur hulk inimesi ka iOSi.

Lisaks sellele, iOS-rakenduse loomine aitab ametiühingul pakkuda paremat kasutajakogemust kõigile liikmetele, olenemata sellest, millist mobiilset operatsioonisüsteemi nad kasutavad.

Selle arenduse edukuse tagamiseks peab arvestama iOSi arendamise spetsiifikaga, mis võib olla erinev Android arendamisest. See sisaldab teadmisi Objective-C või Swift programmeerimiskeeltest, Apple'i arendustööriistade (nagu Xcode) kasutamist, ja iOSi kasutajaliidese suuniste järgimist.

### **7.2 Automaatsete e-kirjade saatmine liikmetele**

Järgmiseks sammuks võiks olla funktsionaalsus, mis võimaldab automaatselt saata kirju ametiühingu liikmetele. Kuna olemas on juba e-posti teenus, oleks seda funktsionaalsust võrdlemisi lihtne rakendada. Funktsionaalsuse kasutamine võiks seisneda selles, et valitakse rühm või osakond ja seejärel saadetakse valitud rühmale automaatselt e-kiri.

## Kasutatud kirjandus

- [1] *UnionWare*. [Võrgumaterjal]. [Kasutatud 15.05.2023]. URL: <https://www.unionware.com/>.
- [2] *eMembership*. [Võrgumaterjal]. [Kasutatud 15.05.2023]. URL: <https://emembership.winmill.com/>.
- [3] *iMIS*. [Võrgumaterjal]. [Kasutatud 15.05.2023]. URL: <https://www.imis.com/union-software>.
- [4] *Union Impact*. [Võrgumaterjal]. [Kasutatud 15.05.2023]. URL: <https://unionimpact.com/>.
- [5] *What is Java technology and why do I need it?*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: [https://www.java.com/en/download/help/whatis\\_java.html](https://www.java.com/en/download/help/whatis_java.html).
- [6] *Gradle Build Tool Features*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: <https://gradle.org/features/>.
- [7] *Why Spring?*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: <https://spring.io/why-spring>.
- [8] *Spring Boot*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: <https://spring.io/projects/spring-boot>.
- [9] *Spring Security*. [Võrgumaterjal]. [Kasutatud 18.04.2023]. URL: <https://docs.spring.io/spring-security/reference/index.html>.
- [10] *Introduction to JSON Web Tokens*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: <https://jwt.io/introduction>.
- [11] *Swagger*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: <https://swagger.io/>.
- [12] *Lombok*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: <https://projectlombok.org/>.
- [13] *MapStruct*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: <https://mapstruct.org/>.
- [14] *Hibernate*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: <https://hibernate.org/>.
- [15] *PostgreSQL*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: <https://www.postgresql.org/about/>.

- [16] *React*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: <https://react.dev/>.
- [17] *Bootstrap*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: <https://getbootstrap.com/>.
- [18] *Android Studio dokumentatsioon*. [Võrgumaterjal]. [Kasutatud 18.04.2023]. URL: <https://developer.android.com/studio>.
- [19] *Best Frontend Frameworks for Web Development in 2023*. [Võrgumaterjal]. [Kasutatud 20.05.2023]. URL: <https://www.simform.com/blog/best-frontend-frameworks/#:~:text=Top%20Frontend%20Frameworks%20of%202023&text=React%20has%20the%20highest%20retention,frameworks%20developers%20wish%20to%20learn..>
- [20] *State of the WordPress Security in 2021*. [Võrgumaterjal]. [Kasutatud 20.05.2023]. URL: <https://patchstack.com/whitepaper/the-state-of-wordpress-security-in-2021/>.
- [21] *Node.js*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: <https://nodejs.org/en/about>.
- [22] *Express*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: <https://expressjs.com/>.
- [23] *Ruby on Rails*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: <https://rubyonrails.org/>.
- [24] *Django*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: <https://www.djangoproject.com/>.
- [25] *Overview of ASP.NET Core*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: <https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-7.0>.
- [26] *GitLab: DevOps Platform*. [Võrgumaterjal]. [Kasutatud 18.04.2023]. URL: <https://about.gitlab.com/>.
- [27] *GitLab Documentation: Continuous Integration/Continuous Deployment*. [Võrgumaterjal]. [Kasutatud 18.05.2023]. URL: <https://docs.gitlab.com/ee/ci/>.
- [28] *Docker*. [Võrgumaterjal]. [Kasutatud 18.04.2023]. URL: <https://docs.docker.com/>.
- [29] *Docker Compose*. [Võrgumaterjal]. [Kasutatud 18.04.2023]. URL: <https://docs.docker.com/compose/>.
- [30] *Nginx*. [Võrgumaterjal]. [Kasutatud 18.04.2023]. URL: <https://nginx.org/en/>.



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

Meie Tarvo Tali, Karl-Gustav Rüütlija Deniel Konstantinov

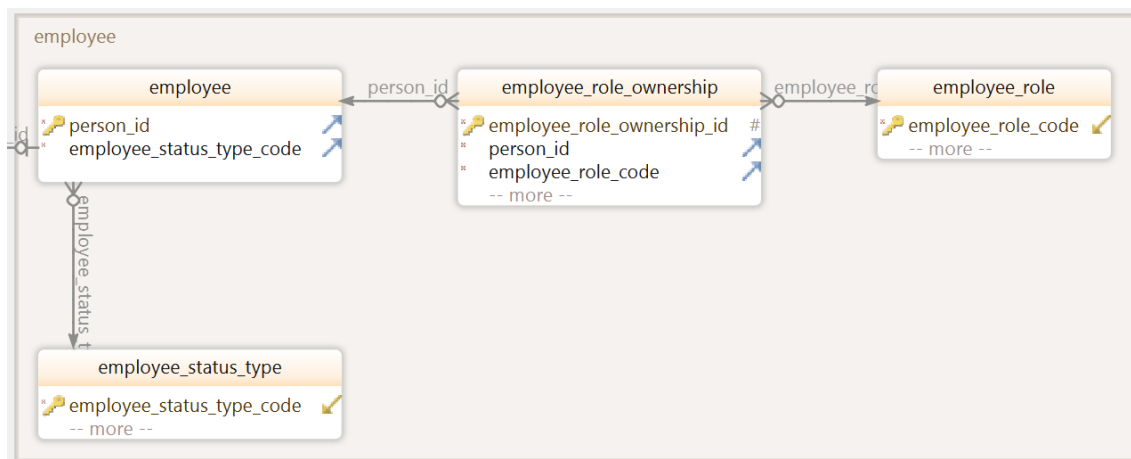
1. Anname Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Development and Implementation of an Information System for Unions”, mille juhendaja on Margarita Matson
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli 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. Oleme teadlikud, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autoritele.
3. Kinnitame, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

23.05.2023

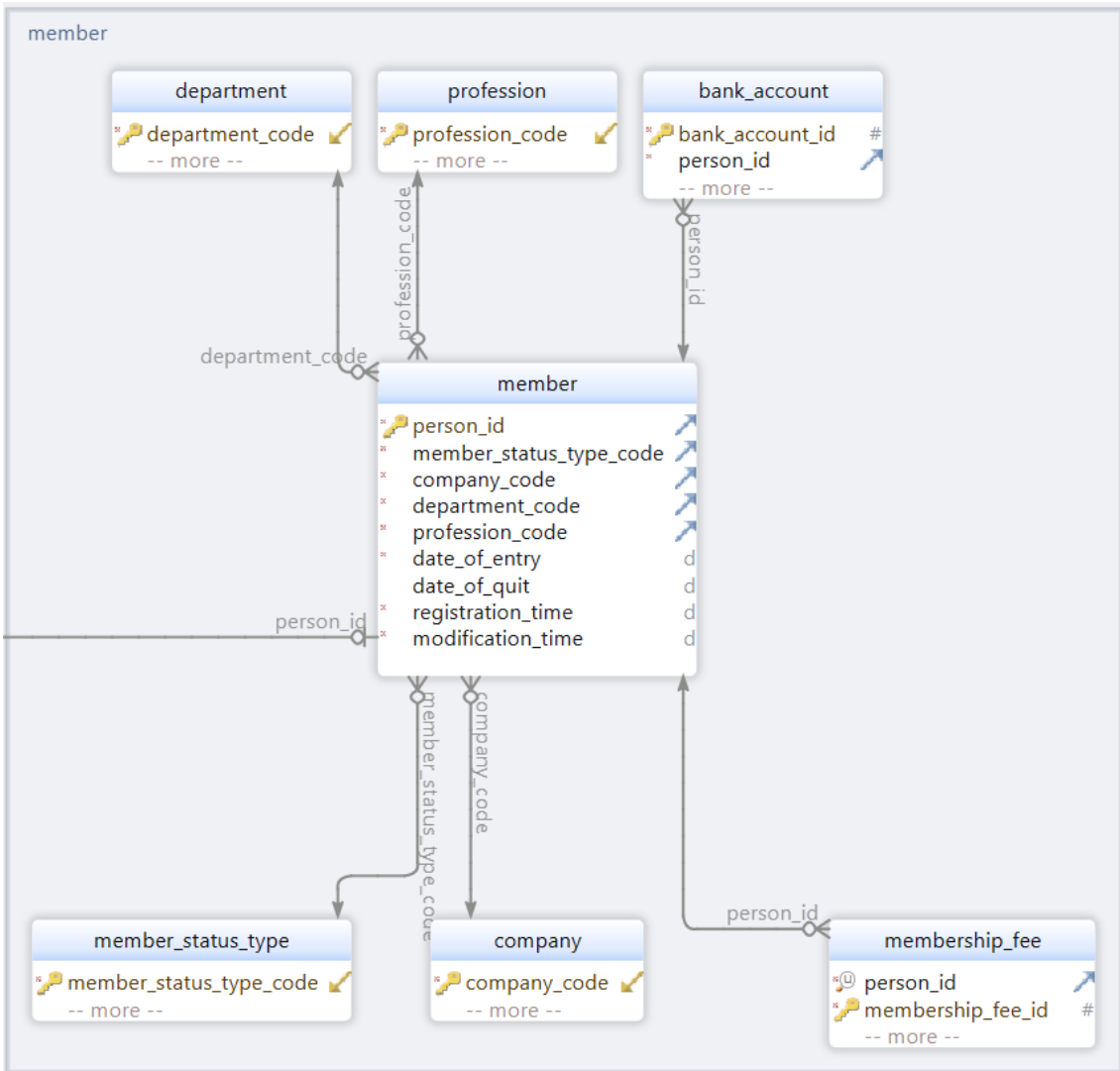
---

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

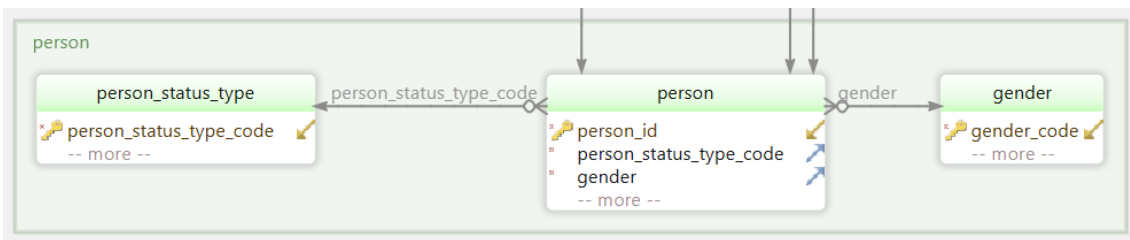
## Lisa 2 – Töötajate rakenduse andmebaasi kontseptuaalne andmemudel



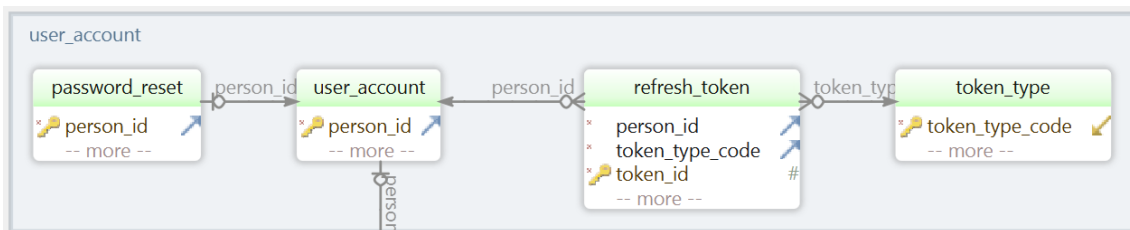
Joonis 1. Töötaja olemi-suhte diagramm.



Joonis 2. Liikme olemi-suhte diagramm.

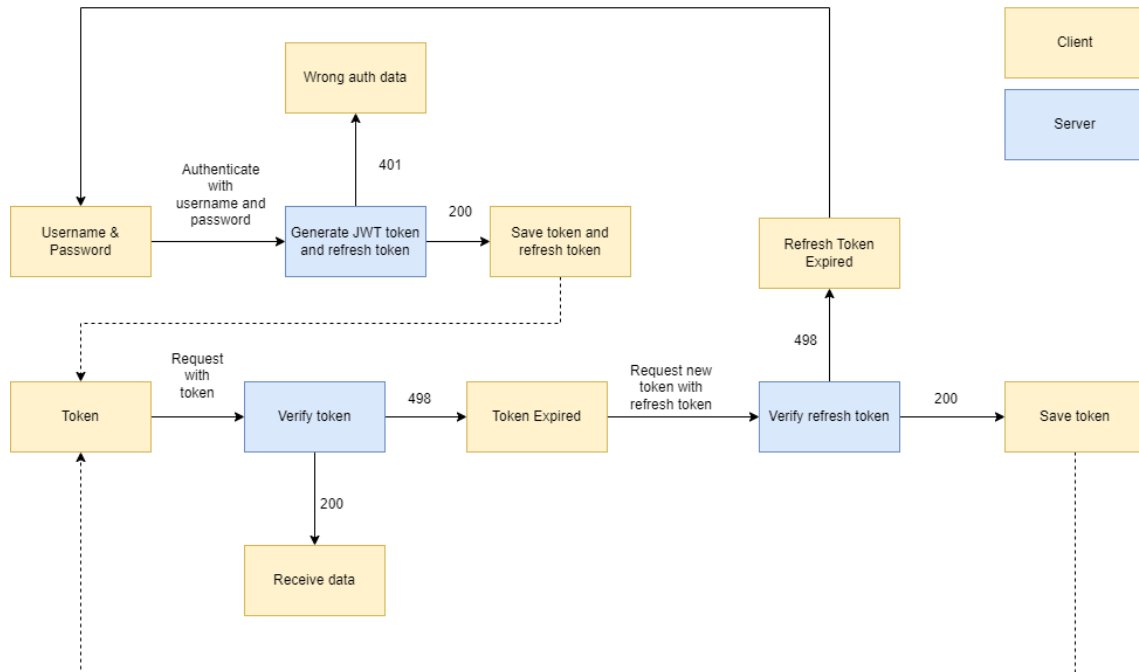


Joonis 3. Isiku olemi-suhte diagramm.



Joonis 4. Kasutajakonto olemi-suhte diagramm.

## Lisa 3 – Autoriseerimise skeem



Joonis 5. Autoriseerimise skeem.

## Lisa 4 – Andmete uude andmebaasi importimise kood

```
db_host = 'localhost'
db_name = 'postgres'
db_user = 'postgres'
db_password = 'postgres'

excel_file = 'filename.xlsx'

df = pd.read_excel(excel_file)

conn = psycopg2.connect(
    host=db_host,
    database=db_name,
    user=db_user,
    password=db_password
)
cursor = conn.cursor()

table_name = 'your_table_name'

sanitized_columns = [col.replace(' ', '_').replace('-', '_') for col in df.columns]
columns = ', '.join(f"{col}" TEXT' for col in sanitized_columns)
create_table_query = f'CREATE TABLE "{table_name}" ({columns})'
cursor.execute(create_table_query)

for index, row in df.iterrows():
    values = [value.strftime('%Y-%m-%d %H:%M:%S') if isinstance(value, datetime) and
              not pd.isnull(value) else None if pd.isnull(value) else value for value in row.values]
    insert_query = f"INSERT INTO {table_name} VALUES ({', '.join(['%s'] * len(values))}"
    cursor.execute(insert_query, tuple(values))

conn.commit()
cursor.close()
conn.close()
```

Joonis 6. Andmete uude andmebaasi importimise kood.