

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Kristo Palo 214804 IADB

Veebirakendus uute töötajate isikuandmete haldamiseks

Bakalaureusetöö

Juhendaja: Meelis Antoi
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: Kristo Palo

15.05.2023

Annotatsioon

Käesoleva bakalaureusetöö eesmärgiks on luua veebirakenduse prototüüp, mille abil saaksid personalitöötajad saata kandidaatidele personaalseid tööpakkumisi ja mis aitaks kandidaatide isikuandmeid paremini hallata ning võimaldaks andmete kogumist teostada efektiivsemalt ja süstematiseeritumalt.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 52 leheküljel, 10 peatükki, 11 joonist, 1 tabelit.

Abstract

**A Web Application for Managing Personal Data of New
Employees**

The goal of this bachelor's thesis is to create a prototype of a web application that would meet specific business requirements of HR specialists involved in conducting new employee onboardings.

The primary objective of this application is to simplify the process of dispatching tailored job offers to job seekers, while also streamlining the collection and management of personal data. By leveraging personalized job offers, the application aims to further enhance the candidate experience and increase the likelihood of successful job placements.

The thesis is in Estonian and contains 52 pages of text, 10 chapters, 11 figures, 1 table.

Lühendite ja mõistete sõnastik

.NET	Microsofti tarkvaraplatvorm.
API	<i>Application Programming Interface</i> – rakendustarkvara liides.
B2B	<i>Business to Business</i> – levinud ärimudel, mille kohaselt müüb üks ettevõtte enda teenuseid teisele ettevõttele.
<i>back end</i>	Tagarakendus, teenusepoolne rakendus.
C#	<i>C Sharp</i> – üldotstarbeline objektorienteeritud programmeerimiskeel, mida kasutatakse laialdaselt mitmesuguste tarkvararakenduste arendamiseks.
CLI	<i>Command Line Interface</i> – käsurea liides.
<i>code-first</i>	Lähenemine, milles luuakse andmebaas programmi koodi järgi.
CSRF	<i>Cross-Site Request Forgery</i> – saidiülene päringu võltsimine.
CSS	<i>Cascading Style Sheets</i> – veebilehtede küljendamisel kasutatav märgistuskeel.
DDD	<i>Domain-Driven Design</i> – tarkvara arendamise lähenemine, mis loob võimekuse ehitada sillad äri vajaduste ja arendatavate tarkvarakomponentide vahele.
DRY	<i>Do not Repeat Yourself</i> – arenduse printsiip, mille kohaselt ei tohiks luua korduvaid arenduse mustreid.
DTO	<i>Data Transfer Object</i> – andmeedastusobjekt. Objekt, mis kannab andmeid erinevate protsesside vahel.
eelprotsessor	Programm või alamprogramm, mis enne suuremat töötlust sooritab mõned töötlussammud.
<i>employee onboarding</i>	Tegevus või protsess uue töötaja integreerimiseks organisatsiooni.
<i>front end</i>	Eesrakendus ehk klientrakendus ehk veebirakenduse kliendipoolne osa.
HTML	<i>Hyper Text Markup Language</i> – veebilehe märgendamise keel.
HTTP	<i>HyperText Transfer Protocol</i> – hüperteksti edastusprotokoll on TCP/IP klient-server protokoll HTML-dokumentide vahetamiseks veebis.
IBAN	<i>International Bank Account Number</i> – rahvusvahelise kontonumbri standardkuju.

IDE	<i>Integrated Development Environment</i> – integreeritud programmeerimiskeskond, mis on rakendustarkvarasse sisse ehitatud.
<i>interceptor</i>	HTTP päringute ja vastuste pealtkuulaja.
JavaScript	Skriptimiskeel, mida kasutatakse peamiselt veebilehtede interaktiivsuse suurendamiseks ja kasutajakogemuse parandamiseks.
JSON	<i>JavaScript Object Notation</i> – lihtsustatud andmevahetusvorming, mis põhineb JavaScripti programmeerimiskeele alamhulgal.
JWT	<i>JSON Web Token</i> – avatud veebitõendid kahe osapoole turvaliseks andmevahetuseks.
<i>lazy loading</i>	Laisk laadimine on andmete või objektide laadimise või initsialiseerimise edasilükkamise praktika seni, kuni neid rakenduse töös päriselt vaja läheb.
LINQ	<i>Language Integrated Query</i> – päringukeel, mis võimaldab rakendusel teha andmebaasi päringuid.
<i>middleware</i>	Vahevara on tarkvara, mis võimaldab omavahel kokku siduda muul juhul mitteühilduvaid tarkvarakomponente või -rakendusi.
MVC	<i>Model-View-Controller</i> ehk mudel-vaade-kontroller on rakendustes kasutatav arhitektuuri disainimuster, mis jaotab rakenduse kolmeks loogiliseks osaks.
obfuskeerimine	Isikuandmete pseudonümiseerimine ehk obfuskeerimine on isikuandmete töötlemine sellisel viisil, et isikuandmeid ei saa enam täiendavat teavet kasutamata seostada konkreetse andmesubjektiga.
OpenAPI	<i>OpenAPI Specification (OAS)</i> – määratleb HTTP veebiliidestele standardse keeleagnostilise kuju, mis võimaldab nii inimestel kui ka masinatel mõista ja kasutada teenuste poolt pakutavaid võimalusi ilma teenuste lähtekoodile, dokumentatsioonile ja võrguliiklusele juurdepääsu omamata.
ORM	<i>Object-Relational Mapping</i> – tarkvaraline komponent, mis vahendab andmeid rakenduse lähtekoodi ja andmebaasi vahel, teisendades objektorienteeritud andmed relatsioonilisteks ja vastupidi.
REST	<i>Representational State Transfer</i> – veebiteenusega suhtlemise liidese arhitektuur.
<i>routing</i>	Marsruutimine on edastamist ootava andmepaketi jaoks sobiva liidese ja järgmise hüppe väljalimine.
SaaS	<i>Software as a Service</i> – pilvepõhine teenus, mille puhul on nii

platvorm kui ka funktsionaalsused hallatud teenuse poolt.

SMTP	<i>Simple Mail Transfer Protocol</i> – lihtne meiliedastusprotokoll. See on üks TCP/IP protokollidest, mis on ette nähtud serveritevaheliseks e-posti sõnumite saatmiseks ja vastuvõtmiseks.
<i>solution</i>	.NET rakenduse projektikaust oma seadistustega.
SPA	<i>Single Page Application</i> – üheleherakendus.
SQL	<i>Structured Query Language</i> – päringukeel, mida kasutatakse relatsioonilistes andmebaasides.
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i> – edastusohje protokollistik internetiprotokollide peal. TCP ja IP protokollid on Interneti protokollikomplektis kaks kõige tähtsamat ja ühtlasi kõige vanemat protokollide.
TypeScript	Tugevalt tüübitud, objektorienteeritud ja kompileeritav programmeerimiskeel, mis põhineb JavaScriptil.
URL	<i>Uniform Resource Locator</i> – igale dokumendile või muule ressursile Internetis vastab oma unikaalne internetiaadress.
XSS	<i>Cross-Site Scripting</i> – koodisüsti meetod veebilehe ründamiseks.

Sisukord

1 Sissejuhatus	13
2 Metoodika.....	14
3 Taustainformatsioon	15
3.1 Fujitsu Estonia AS	15
3.2 Persona.....	15
4 Ülevaade probleemist	16
4.1 Olemasoleva protsessi kirjeldus	16
4.2 Probleemi skoop	16
4.3 Probleemi skoopist lähtuva protsessi kitsaskohad.....	17
4.4 Probleemipüstitus	18
5 Probleemi analüüs	19
5.1 Äriolemid.....	19
5.2 Äriolemite omavahelised seosed	20
5.3 Olemi-suhte diagramm	21
5.4 Süsteemi kasutajad	21
5.5 Kasutajalood	22
5.6 Ärireeglid.....	24
5.7 Isikuandmete töötlemise kord ja seadusandlus.....	25
5.7.1 Mõisted.....	25
5.7.2 Andmekaitse töösuhetes	27
5.7.3 Isikuandmete töötlemise põhimõtted.....	27
5.8 Turul eksisteerivad lahendused	28
5.8.1 Riigitöötaja iseteenindusportaal	28
5.8.2 RecruitLab	29
5.8.3 Google Forms	30
5.8.4 Kokkuvõtte turul eksisteerivatest lahendustest.....	31
5.9 Probleemi analüüsi järeldused	31
6 Tehnoloogiate analüüs	32
6.1 Lähtetingimused ja piirangud	32

6.2 Rakenduse arhitektuur	32
6.3 Klientide haldamine.....	34
6.4 Andmebaasi valik	35
6.5 Tagarakenduse tehnoloogiate valik	35
6.6 Eesrakenduse tehnoloogiate valik	36
6.7 Arenduskeskkonnad ja töövahendid.....	37
6.8 Tehnoloogiate analüüsi järeldused	37
7 Nõuded lahendusele.....	38
7.1 Funktsionaalsed nõuded	38
7.2 Mittefunktsionaalsed nõuded.....	39
8 Lahenduse realiseerimine	41
8.1 Lahenduse nime valik.....	41
8.2 Arendusmetoodika.....	41
8.3 Arendustöö käik.....	42
8.4 Tagarakendus.....	43
8.4.1 Tagarakenduse ülesehitus	43
8.4.2 Veebiliides	45
8.4.3 Äriteenused.....	45
8.4.4 Andmebaas	47
8.5 Eesrakendus	49
8.5.1 Eesrakenduse ülesehitus	49
8.5.2 Suhtlus tagarakendusega	51
8.5.3 Kujundus ja stiilid.....	51
8.6 Turvalisus	52
8.7 Testimine	53
9 Loodud lahendus	56
9.1 Lahenduse kasutamine.....	56
9.1.1 Sisselogimine.....	56
9.1.2 Kandidaatide haldamine	57
9.1.3 Kandidaadi lisamine ja tööpakkumise saatmine.....	57
9.1.4 Tööpakkumise vaatamine ja kinnitamine.....	59
9.1.5 Uue töötaja isikuandmete ankeedi täitmine.....	60
9.2 Lahenduse vastavus lähtetingimustele.....	61
9.3 Kasutajate tagasiside ja võimalused edasiarendusteks	61

9.4 Majandusliku mõju analüüs.....	63
10 Kokkuvõte	64
Kasutatud kirjandus	65
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	69
Lisa 2 – Olemi-suhte diagramm	70

Jooniste loetelu

Joonis 1. Tagarakenduse projektide struktuur Visual Studio kataloogipuus.....	43
Joonis 2. Tagarakenduse projektide omavahelised rakendusesisesed sõltuvused.....	44
Joonis 3. Eesrakenduse projekti struktuur WebStormi kataloogipuus.	50
Joonis 4. Sisselogimise vaade.....	56
Joonis 5. Kandidaatide nimekiri.	57
Joonis 6. Reasisesed tegevused ja toimingud kandidaatidega.....	57
Joonis 7. Uue kandidaadi lisamise modaalaken.	58
Joonis 8. Edukalt lisatud kandidaat.	59
Joonis 9. Kandidaadile saadetud tööpakkumise meil.	59
Joonis 10. Kandidaadile saadetud tööpakkumine.....	60
Joonis 11. Uue töötaja isikuandmete ankeet.....	61

Tabelite loetelu

Tabel 1. Turul eksisteerivate lahenduste võrdlus.	31
--	----

1 Sissejuhatus

Uue töötaja värbamisprotsessi lõppfaasis kulutavad personalitöötajad palju aega sellele, et valituks osutunud kandidaat tööle vormistada. Protsessi ühe osana peab personalitöötaja kandidaadile saatma personaalse tööpakkumise ja seejärel kandidaadi käest koguma ka vajalikud isikuandmed. Kui personaliosakonnas pole juba kasutusel mõnda värbamistarkvara, siis tõenäoliselt toimub andmete kogumine kas e-posti teel, suuliselt või üldsegi paberkujul.

Tööjõuturul valitsev inimeste ja oluliste oskuste puudus sunnib tööandjaid üha enam panustama ka värbamistegevustesse [1]. Kuna enamiku kandidaatide jaoks on värbamisprotsessis osalemine erakordne sündmus, siis ootavad nad kannatamatult informatsiooni ja tagasisidet juba esimesel võimalusel. Tööandja hea maine ning vastastikuse usaldussuhte hoidmiseks on oluline kandidaadile jagada informatsiooni niipea kui võimalik [2]. Uue töötaja tööle vormistamine peaks olema sujuv ja kerge protsess, et kiirendada kogu värbamisprotsessi ja vähendada hea kandidaadi kaotamise riski värbamise lõppfaasis.

Käesoleva lõputöö raames luuakse äriklientidele suunatud veebirakenduse prototüüp, mille abil saaksid personalitöötajad saata kandidaatidele personaalseid tööpakkumisi ja mis aitaks kandidaatide isikuandmeid paremini hallata, võimaldades andmete kogumist teostada efektiivsemalt ja süstematiseeritumalt.

Lahendus on eelkõige mõeldud äriklientidele, kelle värbamismaht on suur ning kellel on juba kasutusel kas ainult personalitarkvara või nii värbamis- kui ka personalitarkvara. Töö raames valmiv lahendus käituks kui võimalik ühenduslüli värbamis- ja personalihaldustarkvara vahel, sidudes need protsessi mõttes üheks tervikuks, kindlustades andmete liikumise ühest süsteemist teise.

2 Metoodika

Käesolev lõputöö koosneb neljast etapist.

Töö esimeses etapis antakse lugejale ülevaade vajalikust taustainformatsioonist. Kirjeldatakse lõputöö autori tööandjat ja senist töökogemust. Sellele järgneb käsitletava probleemi tutvustamine ja olemasoleva protsessi kitsaskohtade kaardistamine. Töö esimene etapp lõpeb lahendatava probleemi sõnastamisega.

Töö teises etapis toimub eelmises etapis püstitatud probleemi põhjalikum analüüs. Kirjeldatakse süsteemis kasutusel olevaid äriolemeid ja nende omavahelisi seoseid. Olemite põhjal koostatakse süsteemi andmemudel. Seejärel kajastatakse juba süsteemis toimetavaid kasutajaid. Igat kasutajagruppi silmas pidades pannakse paika vastava kasutajagrupi kasutajalood. Lisaks fikseeritakse süsteemis kehtivad ärireeglid. Sellele järgneb isikuandmete töötlemise korra ja seadusandluse tutvustus. Ühtlasi antakse lugejale ka ülevaade turul juba eksisteerivatest sarnastest lahendustest. Järgmisena analüüsitakse erinevaid tehnoloogiaid ja lähenemisi, mida võiks võimaliku lahenduse loomise käigus kasutada. Lisaks kaardistatakse ära uuele süsteemile kehtivad lähtetingimused ja piirangud, millega uue lahenduse loomisel peab arvestama. Seejärel pannakse paika võimaliku lahenduse arhitektuur. Arhitektuuri silmas pidades tehakse veel mitmeid teisi tehnoloogilisi valikuid. Teise etapi lõpus määratletakse ära loodavale süsteemile kehtivad nõuded.

Töö kolmandas etapis toimub loodava lahenduse nõutekohane realiseerimine. Lugejale antakse ülevaate arendustöö käigust ja kasutatavast arendusmetoodikast. Kirjeldatakse loodava lahenduse ülesehitust ja toimimise põhimõtteid. Kolmanda etapi lõpuks valmib lahenduse prototüüp, mille abil oleks järgmises etapis võimalik läbi viia kasutajatestimist.

Töö neljandas etapis vaadeldakse loodud lahendust ja tehakse järeldusi. Analüüsitakse seda, mis sai valmis ja korjatakse lõppkasutajate käest tagasisidet. Lisaks tehakse sissevaade rakenduse ärilisse tasuvusse ja viimasena loetletakse võimalusi potentsiaalseteks edasiarendusteks.

3 Taustainformatsioon

Käesoleva töö autor töötab alates 2016. aasta suvest tarkvaraarendajana ettevõttes Fujitsu Estonia AS. 2018. aasta jaanuaris liitus autor sama ettevõtte koosseisu kuuluva tootemeeskonnaga Persona, mis tegeleb samanimelise personalitarkvara pakkumise ja arendamisega.

3.1 Fujitsu Estonia AS

AS Fujitsu Estonia on enam kui 400 töötajaga Eesti ettevõtte, mis kuulub globaalsesse Fujitsu korporatsiooni. Fujitsu Estonia pakub laia valikut IKT teenuseid, lahendusi ja tooteid, toetamaks klientide digitaalset transformatsiooni.

Eestis on kontoreid kokku neli: kaks Tallinnas, üks Tartus ja üks Pärnus.

Ettevõtte on läbi oma eelkäijate Eesti turul tegutsenud alates 1991. aastast. Fujitsu Estonia 2020. aasta majandusaasta käive oli 17 miljonit eurot [3].

3.2 Persona

Persona on personalihalduse, palgaarvestuse ja tööajaplaneerimise B2B (*Business-to-Business*) SaaS-tüüpi (*Software as a Service*) veebilahendus [4].

Persona kuulub Fujitsu Estonia tooteportfelli. Persona tootemeeskonnas töötab tänasel päeval 15 inimest. Käesoleva lõputöö autori roll tarkvaraarendajana hõlmab endas laia spektrit tööülesannetest, alates tarkvaraarendusest kuni kliendisuhtluseni välja.

Persona kontekstis käsitletakse kliendina juriidilist isikut, kellel on Fujitsu Estoniaga sõlmitud teenusleping, kasutamaks Persona personalitarkvara pilvepõhise tarkvarateenusena. Klientide hulgas on nii eraettevõtteid kui ka avaliku sektori asutusi ning organisatsioone, kus on vajadus hallata personaliandmeid. 2017. aastal läbi viidud personalijuhtimise infosüsteemide turu-uuringu kohaselt oli Persona V3 üheks juhtivaks Eesti personalitarkvaraks [5].

4 Ülevaade probleemist

Läbi aja on mitmed Persona kliendid tundnud huvi, kas Persona tooteportfelli on kunagi lisandumas ka *employee onboarding*'u moodul. Viimastel aastatel on vajadus sellise funktsionaalsuse järgi ainult suurenenud, kuna värbamine on üks kõige ajakulukamaid tegevusi personalijuhtimise tegevusnimistus [6].

Kliendid soovivad värbamisprotsessi lõppfaasi ehk töötaja tööle vormistamise protsessi automatiseerida, et seeläbi vähendada enda personalitöötajate koormust ja kiirendada värbamisprotsessi tervikuna.

4.1 Olemasoleva protsessi kirjeldus

Uue töötaja tööle vormistamise protsess algab sellega, et väljavalitud kandidaadile tehakse tööandja poolt ametlik tööpakkumine. Sellele järgneb kandidaadipoolne tööpakkumise vastuvõtmine või mittevastuvõtmine. Tihti toimuvad just selles värbamise etapis ka esimesed tingimuste läbirääkimised kandidaadi ja uue tööandja vahel.

Kui kandidaat on tööpakkumise vastu võtnud, siis peab ta tulevase tööandja palvel enda kohta täitma ära uue töötaja isikuandmete ankeedi. Kandidaadile tehtud tööpakkumine ja täidetud isikuandmete ankeet on eeldusteks, et kandidaadi saaks lisada personaliarvestuse süsteemi ja ette valmistada töösuhet kajastav kahepoolne leping.

Viimasena järgneb lepingu sõlmimine kandidaadi ja tööandja vahel. Kui mõlemad osapooled on lepingu allkirjastanud, loetakse kandidaat tööle vormistatuks ja sellega lõpeb ka värbamisprotsess antud ametikohale.

4.2 Probleemi skoop

Käesoleva lõputöö raames keskendutakse värbamisprotsessi viimase faasi kahele alamprotsessile:

- Kandidaatidele personaalsete tööpakkumiste tegemine.

- Kandidaatide käest isikuandmete kogumine.

4.3 Probleemi skoobist lähtuva protsessi kitsaskohad

Arvestades käesoleva projekti skoopi on olemasoleva protsessi tänased kitsaskohad alljärgnevad.

- Kandidaadile personaalsete tööpakkumiste tegemine:
 - Tööpakkumise koostamine on aeganõudev ja nõuab täpset silma ning ülimat korrektsust.
 - Personalitöötaja peab iga uue kandidaadi puhul alati koostama uue personaalse tööpakkumise.
 - Käsitsi tööpakkumist koostades säilib oht teha andmete sisestamisel inimlikke vigu.
 - Tööpakkumise tegemiseks on personalitöötajal vaja kandidaadiga ühendust võtta kas meili või telefoni teel või paluda kandidaadil tööandja juurde kohale tulla.
- Kandidaadi käest isikuandmete kogumine:
 - Kandidaadi andmed võivad olla kogutud struktureerimata kujul.
 - Kandidaadi andmed võivad olla laiali mitmes andmeallikas.
 - Kandidaadi andmed võivad olla laiali mitme personalitöötaja käes.
 - Kandidaadi andmete sisestamine personaliarvestuse süsteemi on ajamahukas tegevus.
 - Kandidaadi andmete käsitsi sisestamisega personaliarvestuse süsteem võib alati kaasneda inimliku vea tekkimise oht.
 - Kui isikuandmete ankeeti peab täitma paberkujul, siis on kandidaat sunnitud selle täitmiseks tööandja juurde kohale tulema.

- Kui isikuandmeid korjatakse sisendit valideerimata, siis võib personaliarvestuse süsteemi sattuda vigaseid andmeid.

4.4 Probleemipüstitus

1. Persona rakendusel puudub tänasel hetkel *employee onboarding*'u moodul.
2. Kandidaatidele personaalsete tööpakkumiste tegemine on tänasel hetkel Persona klientide hulgas väheautomatiseeritud ja neelab suure osa nende personalitöötajate igapäevatööst, mis omakorda takistab värbamisprotsessi viimase faasiga alustamist.
3. Kandidaatide käest isikuandmete kogumine on tänasel hetkel Persona klientide hulgas väheautomatiseeritud ja see pidurdab kogu kandidaadi tööle vormistamise protsessi kiirust tervikuna.

5 Probleemi analüüs

Peatükis kirjeldatakse esmalt süsteemis kasutusel olevad äriolemeid ja nende omavahelisi seoseid, mille põhjal koostatakse loodavale lahendusele andmemudel. Peale andmemudeli koostamist liigutakse kasutajate juurde ja uuritakse nende käest, kuidas nad sooviksid tulevikus loodavat süsteemi kasutada. Lisaks kaardistatakse ka veel süsteemis kehtivad ärireeglid ja analüüsitakse turul juba eksisteerivaid sarnaseid lahendusi. Peatüki lõpus tehakse probleemi analüüsi käigus saadud teadmiste põhjal kokkuvõtvaid järeldusi.

5.1 Äriolemid

Analüüsi käigus kaardistati alljärgnevad äriolemid. Iga olemi puhul on väljatoodud selle eestikeelne ja ingliskeelne nimetus koos olemi lühikese selgitusega.

- Ametikoht (*Position*) – ettevõtte struktuurne ametikoht.
- Eelleping (*Preliminary Contract*) – kandidaadi ja tööandja vaheline kirjalik/suuline kokkulepe, et kandidaat nõustub talle tehtud tööpakkumises sätestatud tingimustel asuma tööle pakkumise teinud tööandja juures.
- Hädaabikontakt (*Emergency Contact*) – kandidaadi hädaabikontakt on kandidaadile lähedane isik, kellele helistatakse kui kandidaadiga tööl midagi juhtub.
- Hüve (*Benefit*) – mis tahes vormis hüvitis, tasu või muu rahaline või mitterahaline eelis, mida tööandja annab oma töötajatele osana nende üldisest hüvede paketist.
- Isik (*Person*) – füüsiline isik.
- Kandidaat (*Applicant*) – tööandja poolt väljavalitud isik, kes kõnealuse tööandja juures tööle asuda.
- Kontakt (*Contact*) – isiku kontaktandmete kirje.

- Pangakonto (*Bank Account*) – kandidaadi pangakonto informatsioon, mis määrab ära kellele ja kuhu tuleb kanda töötasu.
- Pereliige (*Family Member*) – kandidaadi pereliige või muu lähedane isik.
- Struktuuriüksus (*Structure Unit*) – ettevõtte struktuuriüksus.
- Töökoormus (*Workload*) – ametikoha töökoormus.
- Tööpakkumine (*Job Offer*) – tööandja poolt kandidaadile tehtud formaalne ja viimistletud kutse asuda tööle kõnealuse tööandja juures.
- Töösuhe (*Contract Line*) – kandidaadi töötamise kirje vastava lepingu raames vastaval ametikohal.
- Töötasu (*Remuneration*) – ametikoha eest makstav kokkuleppeline tasu, mis on enamasti kas kuupalk, tunnitasu või lisatasu.
- Ülalpeetav (*Dependant*) – kandidaadi ülalpeetav, kelle ülalpidamise ja eestkoste eest kandidaat isiklikult vastutab.

5.2 Äriolemite omavahelised seosed

Alljärgnevalt on ära kirjeldatud kõikide olemite omavahelised suhted.

- Kõik kandidaadid on isikud.
- Kandidaadil võib olla 0 kuni n kontakti.
- Kandidaadil võib olla 0 kuni n pangakontot.
- Kandidaadil võib olla 0 kuni n hädaabikontakti.
- Kõik hädaabikontaktid on pereliikmed.
- Kõik pereliikmed on isikud.
- Hädaabikontaktil võib olla 0 kuni n kontakti.
- Kandidaadil võib olla 0 kuni n ülalpeetavat.

- Kõik ülalpeetavad on pereliikmed.
- Kandidaat on võib olla seotud 0 kuni n tööpakkumisega.
- Tööpakkumine võib olla seotud 0 kuni n eellepinguga.
- Eelleping võib olla seotud 0 kuni n töötasuga.
- Eelleping võib olla seotud 0 kuni n töösuhtega.
- Töösuhe võib olla seotud 0 kuni n töökoormusega
- Töösuhe on alati seotud ametikohaga.
- Struktuuriüksus võib olla seotud 0 kuni n ametikohaga.
- Struktuuriüksusel võib olla 0 kuni 1 ülem-struktuuriüksust.
- Struktuuriüksusel võib olla 0 kuni n alam-struktuuriüksust.

5.3 Olemi-suhte diagramm

Eelnevalt kirjeldatud äriolemite ja olemite omavaheliste seoste põhjal luuakse olemi-suhte diagramm (Lisa 2). Diagramm on loodud veebipõhise tööriistaga Vertabelo. Olemi-suhte diagrammi koostamisel on arvestatud olemite omavaheliste põlvnemistega ja püütud andmemudeli normaliseerimise abil vähendada sama tähendusega andmete salvestamist mitmes erinevas tabelites. Antud olemi-suhte diagramm on lõpliku andmemudeli lihtsustatud kuju.

5.4 Süsteemi kasutajad

Süsteemil on kahte tüüpi kasutajaid: personalitöötajad ja kandidaadid.

Personalitöötajad on tööandja esindajad, kes on süsteemi otsesed kasutajad. Personalitöötaja jaoks on personalitarkvara üks põhilistest tööriistadest, millega ta igapäevaselt kokku puutub ja enda tööandja personaliprotsesse haldab.

Kandidaadid ehk uued töötajad on eraisikud, kes on avaldanud otsest soovi uue tööandja juures tööd alustada. Kandidaadi staatuses olevad töötajad pole süsteemi igapäevased

kasutajad ja väga paljude jaoks võib see olla esimene ja ehk ka ainuke kokkupuude antud süsteemiga. Süsteemi mõttes ei saa kandidaati veel päris töötajaks lugeda, sest kandidaadi ja vastava asutuse vahel ei pruugi olla sõlmitud rohkem kui vaid suusõnaline kokkulepe. Samuti on võimalik, et kandidaat on enesele teadmata võitnud tööotsingu konkursi aga pole sellest veel ise teadlik. Seetõttu tuleb arvestada, et kandidaadil säilib kuni tööandjaga lepingu sõlmimiseni igasugune õigus ükskõik mis hetkel kandideerimine katkestada ja kogu protsess pooleli jätta.

5.5 Kasutajalood

Kasutajalood on koostatud kliendikohtumistel kogutud info põhjal. Kasutajalugude kajastamisel on keskenduti personalitöötajale ja kandidaadile.

Personalitöötaja kasutajalood:

- Personalitöötajana soovin minimaalse informatsiooni põhjal süsteemi lisada uusi kandidaate, et neile hiljem saata personaalne tööpakkumine.
- Personalitöötajana soovin iga kandidaadi jaoks koostada personaalse tööpakkumise, et kogu pakutava ametikohaga seonduv info selle kaudu viimistletud kujul edasi anda.
- Personalitöötajana soovin, et ma ei peaks tööpakkumise saatmise jaoks kandidaadile kasutajakontot tegema, sest see säästaks minu aega.
- Personalitöötajana soovin kandidaadile saata tööpakkumise elektroonilise teavituse näol, et vähendada personaliosakonnast väljuvate kõnede ja käsitsi kirjutatud emailide hulka.
- Personalitöötajana soovin, et kandidaat saaks tööpakkumises väljatoodud ametikoha kõikide tingimustega ja tööandja poolt pakutavate hüvedega tutvuda juba veebikeskkonnas, et vähendada personaliosakonda sissetulevate infopäringute hulka.
- Personalitöötajana soovin teada, kas kandidaat on personaalse tööpakkumise linki avanud, et vajadusel seda kandidaadile uuesti meelde tuletada.

- Personalitöötajana soovin kandidaatide andmeid muuta, et vajadusel korrastada andmeid ja parandada sisestamise käigus tekkinud vigu.
- Personalitöötajana soovin kandidaate süsteemist kustutada, et vabaneda vigaselt sisestatud või mitteaktuaalsetest kirjetest.
- Personalitöötajana soovin, et kandidaat saaks tööpakkumisega kirjalikult nõustuda veebikeskkonnas, et vähendada personaliosakonda sissetulevate käsitsi kirjutatud kinnituste hulka.
- Personalitöötajana soovin, et peale tööpakkumisega nõustumist suunataks kandidaat automaatselt uue töötaja isikuandmete ankeeti täitma, et ma ei peaks seda kandidaadile eraldi emailiga saatma.
- Personalitöötajana soovin, et kui pole määratud teisiti, siis kandidaat ei tohiks ilma tööpakkumisele nõusolekut andmata pääseda isikuandmete ankeeti täitma, sest see tekitaks hiljem segadust.
- Personalitöötajana soovin, et uue töötaja isikuandmete ankeedi täitmine oleks kandidaadi jaoks piisavalt lihtne ja intuitiivne, et kandidaat saaks selle täitmisega ka iseseisvalt hakkama ega peaks minu poole pöörduma.
- Personalitöötajana soovin, et uue töötaja isikuandmete ankeedi kaudu poleks võimalik saata vääril kujul infot, et vähendada vajadust hiljem vigu parandada.
- Personalitöötajana soovin, et süsteemi kasutamine oleks turvaline, sest ma ei taha, et minu vastutusel olevad andmed satuksid kõrvaliste isikute kätte.

Kandidaadi kasutajalood:

- Kandidaadina soovin, et tööpakkumine saadetakse mulle elektroonsel kujul ja nii pea kui võimalik, sest nii jääb mulle rohkem aega otsuse langetamiseks.
- Kandidaadina soovin, et mulle saadetav tööpakkumine oleks konfidentsiaalne ja oleks saadetud ainult mulle, sest see võib sisalda tundlikke andmeid minu või minu tulevase ametikohta või tööandja kohta.

- Kandidaadina soovin, et mulle saadetav tööpakkumine oleks ülevaatlik, korrektne ja kompaktne, andes mulle võimaluse selle informatsiooni pealt oma otsus langetada, ilma, et ma peaksin selles osas enam kellegi poole pöörduma.
- Kandidaadina soovin, et tööpakkumisele kirjaliku nõusoleku andmine oleks mulle kui kasutajale arusaadav ja nõuaks minult teadlikku kasutajapoolset tegevust ning oleks välistatud, et ma seda kogemata või tähelepanematuses teen.
- Kandidaadina soovin, et peale tööpakkumisele kirjaliku nõusoleku andmist tuleks mulle selle tegevuse kohta ka elektrooniline kinnitus, et minu toiming oli edukas.
- Kandidaadina soovin, et uue töötaja isikuandmete ankeedi täitmine oleks minu jaoks piisavalt lihtne ja mulle jõukohane, et ma saaksin selle täitmisega ka iseseisvalt hakkama ega peaks mõne kolmanda isiku poole pöörduma.
- Kandidaadina soovin, et uue töötaja isikuandmete ankeeti täitmist oleks võimalik suvalisel hetkel pooleli jätta ja hiljem sealt samast kohast jätkata, sest mulle võib ankeedi täitmise ajal mõni muu pakilisem tegevus vahele tulla või pole täitmiseks parasjagu sobiv aeg.
- Kandidaadina soovin, et peale isikuandmete ankeedi täitmist tuleks mulle selle tegevuse kohta ka eraldi elektrooniline kinnitus, et minu minu ankeet sai esitatud.

5.6 Ärireeglid

Järgnevalt kirjeldatakse ärireegleid, mis lihtsustavad äripoole vajadusi arvestades olemite omavahelisi seoseid ja muudavad need päriselule sarnasemaks. Andmemudel jääb aga seejuures samaks, sest vastavaid reegleid saab rakendada ka ainult lahenduse ärioloogika kihis. Nii säilib andmete hoiustamisel siiski parim võimalik kuju ja süsteem jääb piisavalt tulevikukindlaks, kui ärinõuded peaksid ükshetk muutuma.

- Igal kandidaadil peab olema üks kontakti kirje, mille liik on emaili aadress.
- Igal kandidaadil saab korraga olla ainult üks kontakti kirje, mille liik on emaili aadress.

- Igal kandidaadil saab korraga olla ainult üks kontakti kirje, mille liik on elukoha aadress.
- Igal kandidaadil saab korraga olla ainult üks kontakti kirje, mille liik on kontakttelefoni number.
- Kandidaadil saab korraga olla ainult üks pangakonto.
- Kandidaadil saab korraga olla ainult üks hädaabikontakt.
- Hädaabikontaktil saab korraga olla kuni kaks kontakti: kontakttelefoni number ja emaili aadress.
- Kandidaat saab korraga olla seotud ainult ühe tööpakkumisega.
- Tööpakkumise küljes saab korraga olla ainult üks eelleping.
- Eellepingu küljes saab korraga olla ainult üks töösuhe.
- Eellepingu küljes saab korraga olla ainult üks töötasu.
- Töösuhete küljes saab korraga olla ainult üks töökoormus.

5.7 Isikuandmete töötlemise kord ja seadusandlus

Andmekaitse üldmäärus (GDPR) on Euroopa Liidus (EL) kehtiv määrus, mis jõustus 25. mail 2018. GDPR reguleerib üksikisikute isikuandmete töötlemist EL-is ja seab ranged nõuded isikuandmeid töötlevatele organisatsioonidele [7].

Isikuandmete töötlemisega on tegu alati, kui tehakse midagi inimese kohta käivate andmetega. Isikuandmete töötlemine on kasvõi ainuüksi juba isikuandmete kogumine, aga ka näiteks salvestamine, säilitamine, edastamine ja avalikustamine või hävitamine. Isikuandmed on mis tahes andmed tuvastatud või tuvastatava inimese kohta, sõltumata sellest, millisel kujul või millises vormis need on [8].

5.7.1 Mõisted

Isikuandmete kaitse üldmääruse (GDPR) artikkel 4 kohaselt kasutatakse määruses muuhulgas järgmisi mõisteid [9]:

- isikuandmed – igasugune teave tuvastatud või tuvastatava füüsilise isiku ehk andmesubjekti kohta; tuvastatav füüsiline isik on isik, keda saab otseselt või kaudselt tuvastada, eelkõige sellise identifitseerimistunnuse põhjal nagu nimi, isikukood, asukohateave, võrguidentifikaator või selle füüsilise isiku ühe või mitme füüsilise, füsioloogilise, geneetilise, vaimse, majandusliku, kultuurilise või sotsiaalse tunnuse põhjal;
- isikuandmete töötlemine – on isikuandmete või nende kogumitega tehtav automatiseeritud või automatiseerimata toiming või toimingute kogum, nagu kogumine, dokumenteerimine, korrastamine, struktureerimine, säilitamine, kohandamine ja muutmine, päringute tegemine, lugemine, kasutamine, edastamine, levitamise või muul moel kättesaadavaks tegemise teel avalikustamine, ühitamine või ühendamine, piiramine, kustutamine või hävitamine;
- andmete kogum – isikuandmete igasugune korrastatud kogum, millest võib andmeid leida teatavate kriteeriumide põhjal, olenemata sellest, kas kõnealune andmete kogum on funktsionaalsel või geograafilisel põhimõttel tsentraliseeritud, detsentraliseeritud või hajutatud;
- vastutav töötaja – füüsiline või juriidiline isik, avaliku sektori asutus, amet või muu organ, kes üksi või koos teistega määrab kindlaks isikuandmete töötlemise eesmärgid ja vahendid; kui sellise töötlemise eesmärgid ja vahendid on kindlaks määratud liidu või liikmesriigi õigusega, võib vastutava töötaja või tema määramise konkreetset kriteeriumid sätestada liidu või liikmesriigi õiguses;
- volitatud töötaja – füüsiline või juriidiline isik, avaliku sektori asutus, amet või muu organ, kes töötleb isikuandmeid vastutava töötaja nimel;
- andmesubjekti nõusolek – vabatahtlik, konkreetne, teadlik ja ühemõtteline tahteavaldus, millega andmesubjekt kas avalduse vormis või selge nõusolekut väljendava tegevusega nõustub tema kohta käivate isikuandmete töötlemisega;

Käesoleva lõputöö kontekstis on andmete vastutavaks töötlejaks loodava veebirakenduse äriklient ja volitatud töötlejaks selle sama veebirakenduse teenusepakkuja.

5.7.2 Andmekaitse töösuhetes

Tööandja on alati isikuandmete töötleja, sest juba ainuüksi töötaja nimi ja isikukood on isikuandmed. Tööandja on isikuandmete töötleja juba siis, kui ta alles värbab töötajaid, töösuhete kestel, aga ka pärast töölepingu lõppemist [8].

Tööandja ei või lepingueelsetel läbirääkimistel või töölepingu sõlmimist muul viisil ette valmistades, sealhulgas töökuulutuses või töövestlusel, nõuda töölesoovijalt andmeid, mille vastu tal puudub õigustatud huvi (Töölepingu seadus ehk TLS § 11 lg 1). Tööandja õigustatud huvi puudumist eeldatakse eelkõige küsimuste puhul, mis puudutavad ebaproportsionaalselt töölesoovija eraelu või mis ei ole seotud sobivusega pakutavale töökohale (TLS § 11 lg 2). Töötajal on õigus tutvuda tema kohta kogutavate andmetega ja nõuda tegelikkusele mittevastavate andmete kõrvaldamist või parandamist (TLS § 41 lg 1). Tööandja peab tagama töötaja isikuandmete töötlemise vastavalt õigusaktides sätestatule (TLS § 41 lg 2) [10].

5.7.3 Isikuandmete töötlemise põhimõtted

Isikuandmete kaitse seaduse § 14 kohaselt tuleb isikuandmete töötlemisel järgida järgmisi põhimõtteid [8]:

1. seaduslikkus ja õiglus – isikuandmeid töödeldakse seaduslikult ja õiglaselt;
2. eesmärgikohasus – isikuandmeid kogutakse täpselt ja selgelt kindlaksmääratud õiguspärastel eesmärkidel ning neid ei töödelda viisil, mis on nende eesmärkidega vastuolus;
3. kvaliteet – isikuandmed peavad olema piisavad ja asjakohased ning ei tohi olla ülemäärased andmetöötlemise eesmärke arvestades;
4. õigsus – isikuandmed peavad olema õiged ja vajaduse korral ajakohastatud; mõistlike meetmetega tagatakse, et andmetöötlemise eesmärgi seisukohast ebaõiged isikuandmed kustutatakse või parandatakse viivitamata;
5. säilitamine – isikuandmeid säilitatakse kujul, mis võimaldab andmesubjekti tuvastada üksnes seni, kuni see on vajalik selle eesmärgi täitmiseks, milleks isikuandmeid töödeldakse;

6. turvalisus – isikuandmeid töödeldakse viisil, mis tagab nende turvalisuse, sealhulgas kaitseb loata või ebaseadusliku töötlemise eest ning juhusliku kadumise, hävimise või kahjustumise eest, rakendades asjakohaseid tehnilisi või korralduslikke meetmeid;

Kuna personalitarkvara sisaldab isikuandmeid, peab see ka vastama GDPR eeskirjadele. GDPR-i kohaselt peavad organisatsioonid (vastutavad töötlejad) enne isikuandmete kogumist ja töötlemist saama üksikisikutelt selgesõnalise nõusoleku.

Kui isikuandmete töötlemine põhineb andmesubjekti nõusolekul, peab vastutaval töötlejal olema võimalik tõendada, et andmesubjekt on nõustunud enda isikuandmete töötlemisega. Kui andmesubjekt annab nõusoleku kirjaliku kinnituse, mis puudutab ka muid küsimusi, esitatakse nõusoleku taotlus viisil, mis on muudest küsimustest selgelt eristatav, ning arusaadaval ja lihtsasti kättesaadaval kujul, kasutades selget ja lihtsat keelt. Andmesubjektil on õigus oma nõusolek igal ajal tagasi võtta. Nõusoleku tagasivõtmine peab olema sama lihtne nagu oli selle andmine (GDPR artikkel 7) [11].

Personalitarkvara pakkujad (volitatud töötlejad) peavad tagama, et nende tarkvara on GDPR-iga kooskõlas ning pakkuma vajalikke tööriistu ja funktsioone, mis aitavad organisatsioonidel GDPR-i eeskirju järgida. See võib hõlmata andmete krüpteerimist, varundamist ja taastamist, andmetele juurdepääsu reguleerimist ja võimalust andmeid nõude peale kustutada või pseudonümiseerida. Nii vastutavad kui ka volitatud töötlejad on kohustatud rakendama asjakohaseid tehnilisi ja korralduslikke meetmeid isikuandmete turvalisuse tagamiseks (GDPR artikkel 32) [12].

5.8 Turul eksisteerivad lahendused

Järgnevalt tutvustatakse mõningaid juba praegu Eesti turul eksisteerivaid tarkvaralahendusi, mille abil on võimalik kas osaliselt või täielikult lahendada eelmises peatükis püstitatud probleeme. Kirjelduste koostamisel on tuginetud Internetis leiduval avalikul informatsioonil.

5.8.1 Riigitöötaja iseteenindusportaal

Riigitöötaja iseteenindusportaal (RTIP) on keskkond, mis automatiseerib info liikumise asutustest riigi finants-, personali- ja palgaarvestuse süsteemi SAP. Portaal koosneb

kuuest moodulist: isikuandmed, puhkused, lähetused, koolitused, majanduskulud ja varad [13].

Asutuse personalitöötaja saab RTIP isikuandmete mooduli kaudu vormistada töösuhteid uute töötajatega. Seoses uue töötaja tööle vormistamisega RTIPis tuleb täita isikuandmete ankeet, mis koosneb asutuse poolsest osast ning kandidaadi täidetavast osast. Et kandidaat teaks tulla portaali täitma isikuandmete ankeeti ja aktsepteerida tööpakkumist, tuleb talle saata e-mailiga kutse. Kandidaadil on võimalik siseneda iseteenindusportaali, täita enda kohta vajalikud andmed ning saata täidetud ankeet asutusse tagasi, personalitöötajale kontrollimiseks ja andmete registrisse kandmiseks [14].

Riigitöötaja iseteenindusportaal RTIP oli käesoleva töö autorile kõige suuremaks eeskujuks. Teadaoleva informatsiooni põhjal pole aga antud tarkvara kahjuks mõeldud kommertskasutuseks, vaid ainult riigiasutusele. See inspireeris töö autorit looma sarnast lahendust ka eraettevõtetele.

5.8.2 RecruitLab

RecruitLab on värbajakeskne tööandja turunduse ja kandidaatide haldamise tarkvara [15]. RecruitLabi veebirakendust arendab ja pakub Eesti ettevõtte Recruitment Software OÜ.

RecruitLab automatiseerib värbamises kõike, mida automatiseerida on võimalik ja annab värbajatele võimaluse tegeleda enam väärtust loovate ja tavaliselt ka huvitavamate tegevustega. RecruitLab hoiab kokku värbaja väärtuslikku tööaega ja aitab tagada positiivse kandideerimiskogemuse kandidaatidele. Värbajad ei peaks tegelema tegevustega, mis on rutiinsed, ajamahukad ja mida saab lihtsa vaevaga automatiseerida. Kandidaadid teisalt ei tohiks jääda infosulgu, mis on üks peamistest negatiivse kandidaadikogemuse põhjustest. RecruitLabi meeskond usub, et värbamisprotsessi on võimalik tehnoloogia abil muuta mõlemale osapoolle mõnusamaks ja tulemuslikumaks [16].

RecruitLab pakub mitmeid funktsionaalsusi: kandidaatide haldamine; töökuulutuste ja karjäärilehtede loomine; värbamise mõõdikud ja statistika; paindlikud kandideerimisvormid; sotsiaalmeedia värbamine; kontroll ja läbipaistvus enda ettevõtte värbamisprotsessi üle [17].

RecruitLab küündib esile mitmel põhjusel. Esiteks sellepärast, et nad pakuvad samuti äriklientidele suunatud terviklikku pilvepõhist veebilahendust. Teiseks on nende rakenduses võimalik koostada paindlikke kandideerimisvorme ja saata väljavalitud kandidaatidele tööpakkumisi. Lisaks pakub RecruitLab ka integratsioone mitmete välise personalitarkvaradega. Uut lahendust luues arvestatakse sellega, et seda oleks hiljem võimalik lihtsasti liidestada erinevate värbamistarkvaradega.

5.8.3 Google Forms

Google Forms ehk Google'i vormid on tasuta veebipõhine rakendus, mis võimaldab kasutajatel koostada ja jagada küsitlusi ja läbi viia mitmesuguseid uuringuid.

Google Forms pakub lihtsat ja intuitiivset liidest kohandatud uuringute, küsimustike, tagasisidevormide, registreerimisvormide ja muu loomiseks. Kasutajad saavad valida mitmesuguste küsimuste tüüpide hulgast, näiteks valikvastustega, märkeruudu, rippmenüü, lühikese vastuse ja muu hulgast. Samuti võimaldab see kasutajatel vormile pilte ja videoid lisada. Kui vorm on loodud, saavad kasutajad seda lingi kaudu teistega jagada. Vastuseid kogutakse automaatselt ja neid saab reaalselt vaadata.

Google'i vormidel on palju praktilisi rakendusi, nagu turu-uuringute läbiviimine, klientide tagasiside kogumine, sündmuste jaoks vastuste kogumine, hariduslikel eesmärkidel viktoriinide loomine ja isegi töötajate kaasamiseks [18]. Ühtlasi on võimalik Google'i vormide abil muuta ka uute töötajate liitumisprotsess sujuvamaks ja tõhusamaks, kasutades seda kandidaatide käest vajalike isikuandmete kogumiseks, elektroonse isikuandmete ankeedi näol. Personalitöötajad saavad ankeetide andmeid eksportida arvutustabelitesse edasiseks analüüsiks ja visualiseerimiseks.

Google Formsi suur kasutusmugavus ja mitmekülgsus muudavad selle populaarseks tööriistaks igas suuruses ettevõtetele ja organisatsioonidele. Paraku on aga Google'i vormide integreerimine välise personalitarkvaraga pigem keeruline ja seal puudub võimalus saata kandidaatidele personaalseid tööpakkumisi.

5.8.4 Kokkuvõte turul eksisteerivatest lahendustest

Tabelis 1 on kujutatud eelpool kirjeldatud turul eksisteerivate lahenduste võrdlus.

Tabel 1. Turul eksisteerivate lahenduste võrdlus.

Omadus	RTIP	RecruitLab	Google Forms
Tasuline	Info puudub	Jah	Ei
Mõeldud kasutamiseks eraettevõtetele	Ei	Jah	Jah
Personaalsete töopakumiste koostamine ja saatmine	Jah	Jah	Ei
Dünaamiline isikuandmete ankeet	Info puudub	Teatud määrani seadistatav	Ei
Isikuandmete ankeedi andmeväljade ärioloogiline valideerimine	Jah	Info puudub	Ei
Kandidaadi poolt esitatud andmete muutmine	Jah	Jah	Ei
Liidestatav välise personalihaldustarkvaraga.	Info puudub	Jah	Nõuab eriarendust

5.9 Probleemi analüüsi järeldused

Kuna turul praegu sobivat lahendust ei leidu, siis oleks kõige otstarbekam luua uus terviklik veebirakendus, mille abil saaksid personalitöötajad kandidaatidele saata töopakumisi ning koguda nende isikuandmeid.

6 Tehnoloogiate analüüs

Järgnevalt fikseeritakse loodavale lahendusele kehtivad lähtetingimused ja piirangud. Seejärel antakse lugejale ülevaade mõningatest võimalikest lähenemistest, tehnoloogiatest ja tööriistadest, mida saaks lahenduse loomise käigus kasutada. Nende hulgast valitakse välja kõige sobilikumad ja peatüki lõpus esitatakse ettepanekud loodava süsteemi tehnoloogiliste valikute osas.

6.1 Lähtetingimused ja piirangud

Uue lahenduse loomisel peab arvesse võtma alljärgnevaid tehnoloogilisi ja ärilisi piiranguid:

- Loodav lahendus peab arvestama Persona tänase süsteemi eripäradega ja püüdma peegeldada tänase süsteemi ülesehitust ja parimaid praktikaid ning võimalusel ka kasutatud tehnoloogiaid. See aitab hoida kasutatavate praktikate, disainimustrite ja tehnoloogiatega hulga ning seeläbi ka kogu Persona tooteportfelli ökosüsteemi entroopia kontrolli all [19]. Lisaks loob see eelduse, et kui prototüüp on edukas, siis saab prototüübi ehitamise käigus tehtud tööd maksimaalselt ära kasutada ja selle lihtsamini Persona tooteportfelli integreerida.
- Loodav lahendus ei tohi toetuda tänasele Persona rakendusele ega mõjutada selle käideldavust. Kogu uue süsteemi arendusprotsess tuleks hoida Persona omast lahus ja kõnealust uut lahendust peaks käsitlema iseseisva süsteemina.
- Loodav lahendus peab omama avalikku veebiliidest, mille abil oleks see võimalik liidestada tänase Persona rakendusega.

6.2 Rakenduse arhitektuur

Uue veebirakenduse kavandamisel mängib üliolulist rolli rakendusele õige arhitektuuri valimine. Kõige populaarsema või uuema turul leiduva tehnoloogia valimine ei pruugi alati tähendada seda, et see ka parima tulemuse toob. Kõige sobivama lähenemise

valimine annab aga tõestatud lahendused mitmetele praegustele ja tulevikus ilmnevatele probleemidele [20]. Kõige optimaalsema arhitektuurini jõudmiseks on soovitatav kasutada mõnda üldtunnustatud tarkvara arhitektuurimustrit.

Tarkvara arhitektuurimuster on üldine korduvkasutatav lahendus tarkvara arhitektuuri kujundamisel sageli esinevale probleemile. See on tõestatud ja usaldusväärne viis tarkvarasüsteemide disainimiseks ja struktureerimiseks, muutes need lihtsasti mõistetavaks, muudetavaks ja hooldatavaks.

Järgnevalt tutvustatakse kolme levinud arhitektuurimustrit, mis kõik hõlmavad väga erinevaid kasutusjuhtumeid. Nendeks on monoliit, kihiline monoliit ja mikroteenused. Just need mustrid näivad olevat leidnud kõige sagedasemat kasutust tänapäeva tarkvarakogukondades [21]. Kõigil kolmel esineb teatud eeliseid ja puuduseid, mistõttu tuleb valiku tegemisel arvestada mitme asjaoluga, näiteks rakenduse suuruse ja selle keerukusega, skaleeritavuse nõuetega, arendusmeeskonna suurusega ning rakenduse arendus- ja hoolduskuludega.

- Monoliitsed rakendused on mõeldud mitmete seotud ülesannete terviklikuks lahendamiseks [22]. Monoliitse arhitektuuriga rakendused võivad olla keerulise ülesehitusega, kuna need on loodud olema iseseisvad: rakenduse juurutamiseks või kompileerimiseks peavad kõik selle jaoks vajalikud komponendid alati rakenduse juures olema. Monoliitsetest rakendustest võib mõelda kui traditsioonilisest viisist luua veebirakendusi [23]. Selline lähenemine sobib hästi väikese ja keskmise suurusega, lihtsa äriloogikaga rakenduste jaoks, sest seda on kiire juurutada ning lihtne hooldada.
- Kihiline arhitektuur jagab rakenduse mitmeks loogiliseks kihiks: esitluskiht, äriloogikakiht ja juurdepääs andmetele. Kihid on viis vastutuste eraldamiseks ja sõltuvuste haldamiseks. Iga kiht vastutab kindla funktsiooni eest ja suhtleb ainult vahetult selle all oleva kihiga. Kõrgem kiht saab kasutada madalama kihi teenuseid, kuid mitte vastupidi [15]. Kihiline arhitektuur sobib hästi keskmise suurusega rakenduste jaoks, kus on oluline eristada rakenduse erinevaid osasid. See võimaldab suuremat skaleeritavust ja lihtsamat hooldatavust.
- Mikroteenuste arhitektuur on hajutatud süsteem, mis jaotab rakenduse väikesteks sõltumatuteks teenusteks. Iga teenus vastutab konkreetse funktsiooni eest, töötab

eraldi protsessina ja suhtleb teiste teenustega veebiliidese või sõnumside kaudu. See lähenemine sobib hästi suurte ja keerukate rakenduste jaoks, kus rakenduse komponentidel on erinevad skaleeritavuse, kättesaadavuse ja tõrketaluvuse nõuded. Mikroteenuste arhitektuur võimaldab teenuste iseseisvat arendamist, juurutamist ja skaleerimist, mis toob kaasa kiiremad arendustsüklid ja suurema paindlikkuse.

Käesoleva lõputöö kontekstis on monoliidi ehitamine pigem välistatud, kuna see lukustaks loodava rakenduse kindlate tehnoloogiliste ja äriliste valikute külge, mistõttu võib seda hiljem olla raske ülejäänud Personaga integreerida. Mikroteenuste kasutamine ei tundu antud juhul samuti mõistlik, kuna lahendatav probleem ei ole nii suur, et selle täiendav tükeldamine väiksemateks teenusteks ennast ära tasuks. Leitakse, et uue süsteemi loomisel oleks kõige otstarbekam kasutada klassikalist kolmekihilist arhitektuuri (esitluskiht, ärioloogikiht, andmekiht), sest see on enda sobivust Persona rakenduse juures juba tõestanud ja töö autoril on sellega head kogemused.

6.3 Klientide haldamine

Klientide haldamiseks kasutatakse Personas mitmikkliendi arhitektuuri (*Multitenancy*). Mitmikkliendi arhitektuur tähendab seda, et kõikide klientide andmed asuvad ühes andmebaasis koos ja kõik kliendid kasutavad sama veebirakendust. Andmebaas on klientide vahel jagatud ja kliendid on üksteisest loogiliselt eraldatud [24]. Andmebaasis kasutatakse klientide eristamiseks unikaalseid kliendi identifikaatoreid.

Mitmikkliendi arhitektuuri eelised:

- Serverite majutusteenus on kuluefektiivne, sest serverite halduskulu jaguneb kaudselt kõikide klientide vahel laiali.
- Veebirakenduse ja andmebaasi hooldus ning haldamine on lihtne, sest nii veebirakendusest kui ka andmebaasist eksisteerib ainult üks eksemplar.

Mitmikkliendi arhitektuuri puudused:

- Pikad andmebaasi transaktsioonid ja aeglased SQL päringud võivad olulisel määral mõjutada kõikide klientide rakenduse töökiirust.

- Andmete taastamine andmebaasi varukoopiast võib olla keeruline protsess, sest kui taastada on vaja ainult ühe kliendi andmete varasem seis, siis see tähendab, et ülejäänud klientide andmed ei tohi selle tegevuse käigus muutuda.

Lahenduse loomisel kasutatakse mitmikkliendi arhitektuuri, sest see on ennast tänase Persona rakenduse juures juba aastate jooksul tõestanud ja kõnealuse lahenduse töökoormus hakkab vähemalt esialgu olema pigem väike ning praegu ei ole ette näha, et see peaks lähitulevikus hüppeliselt suurenema.

6.4 Andmebaasi valik

Persona tänane andmekiht kasutab andmebaasina Microsoft SQL Serverit. Töö autoril on sellega eelnev kogemus.

Kui loodav lahendus läbib prototüübi faasi edukalt, siis peab selle andmebaas olema paigaldatav olemasolevasse SQL Serveri andmebaasimootoris, tänase Persona andmebaasi kõrvale.

Kui kunagi peaks tekkima vajadus tänane Persona rakendus ja käesoleva lõputöö käigus loodav lahendus üheks rakenduseks kokku liita, siis on võimalik loodava lahenduse andmebaasi andmetabelid siirdada otse Persona tänasesse andmebaasi. See tagab loodava lahenduse andmekihile teatava tulevikukindluse.

Ehkki valikus on veel ka andmebaasimootorid PostgreSQL või MySQL, leiab töö autor, et uut lahendust luues oleks siiski kõige mõistlikum jääda senise andmebaasi teenusepakkuja juurde, kuna Microsoft SQL Serveri kasutuslitsents on juba ostetud ja pigem on andmebaas andmemahu mõttes alakoormatud.

6.5 Tagarakenduse tehnoloogiate valik

Persona tänane ärioloogikakiht on programmeeritud programmeerimiskeeles C# ja on ehitatud veebiraamistikule .NET Core. Tänapäevane Persona tagarakendus (*back end*) on majutatud Microsofti veebiserverisse IIS. Töö autoril on kõigi kolmega eelnev kogemus.

Kui loodav lahendus läbib prototüübi faasi edukalt, siis peab selle tagarakendus olema paigaldatav olemasolevasse IIS-i veebiserverisse, tänase Persona serverirakenduse kõrvale.

Kui kunagi peaks tekkima vajadus tänane Persona rakendus ja käesoleva lõputöö käigus loodav lahendus üheks rakenduseks kokku liita, siis on võimalik loodava lahenduse ärikiht siirdada otse Persona tänasesse ärikihti. See tagab loodava lahenduse ärikihile teatava tulevikukindluse.

Ehkki valikus on veel ka serveripoolse veebirakenduse raamistikud Java Spring Boot, Express.js ja Django, leiab töö autor, et uut lahendust luues oleks siiski kõige parem valik jääda senise programmeerimiskeele ja veebiraamistiku juurde. Uue täiendava tehnoloogilise platvormi kasutuselevõtmine ei tundu ratsionaalselt põhjendatav, kuna kogu mõeldav funktsionaalsus on ka juba praegusel platvormil olemas. Ka rakenduse halduskulude mõttes ei ole vahet, sest .NET Core veebirakendus on iseenesest paigaldatav ka Linux operatsioonisüsteemi.

6.6 Eesrakenduse tehnoloogiate valik

Persona tänane esitluskiht on programmeeritud TypeScriptis ja on ehitatud veebiraamistikule Angular. Klientrakendus (*front end*) on majutatud nginx-i veebiserverisse. Töö autoril on Angulariga vähe kogemust.

Kui loodav lahendus läbib prototüübi faasi edukalt, siis peab selle esitluskiht olema paigaldatav olemasolevasse nginx-i veebiserverisse, tänase Persona klientrakenduse kõrvale.

Kui kunagi peaks tekkima vajadus tänane Persona rakendus ja käesoleva lõputöö käigus loodav lahendus üheks rakenduseks kokku liita, siis on võimalik loodava lahenduse esitluskiht siirdada otse Persona tänasesse esitluskihti. See tagab loodava lahenduse esitluskihile teatava tulevikukindluse.

Ehkki valikus on veel ka React ja Vue.js, leiab töö autor, et Persona kõrvale uut rakendust luues oleks siiski kõige sobilikum jääda senise klientrakenduse arendusplatvormi ja teekide juurde, kuna see annaks võimaluse luua korduvkasutatavaid komponente ja neid hiljem mõlema rakenduse juures taaskasutada.

6.7 Arenduskeskkonnad ja töövahendid

Tagarakenduse arendamiseks kasutati Microsofti IDE-d (*Integrated development environment*) Visual Studio 2022 Enterprise, sest autoril on selle tootega kõige suurem kogemus.

Eesrakenduse arendamiseks kasutati JetBrainsi loodud IDE-d WebStorm. WebStorm on käesoleva töö autori arvates kõige intelligentsem JavaScripti arenduskeskkond ning seda on lihtne ja mugav kasutada.

Andmebaasi haldamiseks ja selles kontrollpäringute tegemiseks kasutati SQL Server Management Studiot, sest on SQL Serveri haldamiseks mõeldud vaikerakendus ja autoril on sellega kõige suurem kogemus.

Enda tööaja planeerimiseks ja jälgimiseks kasutab töö autor veebirakendust Toggl Track.

Serverirakenduse veebiliidese käsitsi testimiseks kasutati rakendusse sisseehitatud tööriista Swagger ja välist veebiteenuste testimiseks mõeldud tööriista Postman.

Meiliteavituste saatmise ja kättesaamise kontrollimiseks kasutati veebirakendust Ethereal.

6.8 Tehnoloogiate analüüsi järeldused

Käesoleva töö käigus loodav lahendus peaks olema iseseisev, äriklientidele suunatud, kolmekihilise ja mitmikkliendi arhitektuuriga veebirakenduse prototüüp. Rakendus ise peaks moodustama terviku Microsoft SQL Serveri andmebaasist, ASP.NET Core platvormile kirjutatud serveripoolsest tagarakendusest ja Angulari veebiraamistikus kirjutatud kliendipoolsest eesrakendusest.

7 Nõuded lahendusele

Nõuete määramisel lähtuti kaardistatud tehnoloogilistest piirangutest, koostatud kasutajalugudest ja kirjeldatud ärireeglitest.

7.1 Funktsionaalsed nõuded

Personalispetsialisti funktsionaalsed nõuded:

- Personalitöötaja peab saama rakendusse sisse logida, kasutades autentimiseks kasutajanime ja parooli.
- Personalitöötaja ei tohi saada rakendust kasutada ilma ennast eelnevalt autentimata.
- Personalitöötaja ei tohi näha teiste asutuste andmeid.
- Personalitöötaja peab nägema kõikide ametikohtade loetelu.
- Personalitöötaja peab saama süsteemi lisada uusi kandidaate.
- Personalitöötaja peab saama igale kandidaadile koostada personaalse tööpakkumise.
- Personalitöötaja peab saama kandidaadi meili peale saata personaalse tööpakkumise.
- Personalitöötaja peab saama kontrollida, kas kandidaat on personaalse tööpakkumise avanud.
- Personalitöötaja peab saama kandidaadi kõiki andmeid muuta.
- Personalitöötaja peab saama kandidaadi kõiki andmeid kustutada.

Kandidaadi funktsionaalsed nõuded:

- Kandidaat peab saama rakendust kasutada ilma autentimata.

- Kandidaat peab saama vaadata talle saadetud tööpakkumist.
- Kandidaat ei tohi näha teiste kandidaatide tööpakkumisi.
- Kandidaat peab saama kinnitada talle saadetud tööpakkumise.
- Kandidaat ei tohi saada kinnitada teiste kandidaatide tööpakkumisi.
- Kandidaat peab saama täita talle saadetud uue töötaja isikuandmete ankeeti.
- Kandidaat ei tohi saada täita teiste kandidaatide uue töötaja isikuandmete ankeete.

7.2 Mittefunktsionaalsed nõuded

Üldised mittefunktsionaalsed nõuded rakendusele:

- Rakendus peab olema iseseisev pilvepõhine SaaS-tüüpi veebirakendus.
- Rakendus peab kasutama kolmekihilist arhitektuuri MVC (*Model-View-Controller*).
- Rakendus peab klientide haldamiseks kasutama mitmikkliendi (Multitenancy) arhitektuuri.
- Rakendus peab vastama kõikidele Eesti Vabariigis kehtivatele isikuandmete töötlemist puudutavatele seadustele ja järgima parimaid põhimõtteid.
- Rakendus peab toetama veebibrausereid Google Chrome, Mozilla Firefox ja Edge.

Mittefunktsionaalsed nõuded tagarakendusele:

- Tagarakendus peab kasutama andmebaasi Microsoft SQL Server.
- Tagarakendus peab olema kirjutatud programmeerimiskeeles C#.
- Tagarakendus peab olema ehitatud veebiplatvormile ASP.NET Core.
- Rakendus peab pakkuma avalikke *RESTful* (*Representational State Transfer*) veebiteenuseid.

- Tagarakendus peab läbi avaliku veebiliidese (API) olema liidestatav välise personalihaldustarkvaraga Persona.
- Tagarakenduse kaudu peab olema võimalik saata kasutajatele meiliteavitusi.

Mittefunktsioonalsed nõuded eesrakendusele:

- Eesrakendus peab olema kirjutatud programmeerimiskeeles TypeScript.
- Eesrakendus peab olema ehitatud veebiplatvormile Angular.
- Loodav lahendus peab olema kasutatav eesti keeles.

Mittefunktsioonalsed nõuded kasutuskogemusele ja kasutajaliidesele:

- Rakenduse kasutamine peab kasutaja jaoks olema lihtne ja intuitiivne.
- Rakenduse kasutajaliides peab olema moodne ja visuaalselt kena.
- Rakenduse kasutajaliides peab toetama töölauavaadet.
- Rakenduse kasutajaliides peab toetama mobiilivaadet.

8 Lahenduse realiseerimine

Eelmistes peatükkides püstitatud probleemi otsustati lahendada uue iseseisva veebirakenduse prototüübiga.

8.1 Lahenduse nime valik

Rakenduse nimeks saab Sulefa. Nimi on tuletatud laste mängu „Kivi-paber-käärid“ Lõuna-Eesti-pärasest nimest „Su-Je-Fa“. Rakenduse nimi „Sulefa“ tähistab lihtsust ja mängulisust. Sulefa lähtub põhimõttest, et ka äritarkvara kasutuselevõtt ja kasutamine peaks kasutaja jaoks olema sama sujuv ja intuitiivne kui lapsemäng.

8.2 Arendusmetoodika

Ehkki rakenduse arendamisega tegeleti iseseisvalt ja omas tempos, rakendati siiski teatud agiilse tarkvaraarenduse põhimõtteid. Analüüsietapis kogutud kasutajalugude, ärireeglite ja nõuete põhjal koostati arendustööde nimekiri. Arendustööde nimekirja kirjeid nimetatakse kanneteks. Igal kandel on vähemalt pealkiri ja kirjeldus.

Kui esialgne arendustööde nimekiri on valmis, algas tööde planeerimine. Selleks tuvastati esmalt kannete omavahelised sõltuvused ning seejärel pandi paika tööde teostamise järjekord, mille alusel jagati kanded arendustöö iteratsioonidesse ehk sprintidesse. Lähtuti põhimõttest, et igal suvalisel ajahetkel peaks alati olema planeeritud vähemalt käimasolev ja tulevane sprint.

Sprintide koostamisel arvestati seda, et 2/3 sprindi mahust peaks jääma uute arenduste jaoks ja 1/3 sprindi mahust peaks jääma eelmiste sprintide käigus tekkinud vigade parandamise jaoks. Niiviisi planeerides on sprintide arvatavad lõpptähtjad realistlikumad ja seeläbi on ka korraga üleskerkinud vigade hulk alati kontrolli all. Selline dünaamilisus annab suurema paindlikkuse arendustööde planeerimisel ja on tagatud, et alati tegeletakse just nende töödega, mis on antud hetkel kõige olulisemad.

Sprindid koosnevad nii arendustegevusest kui ka testimisest. Iga sprindi lõpuks peab loodav rakendus säilitama oma terviklikkuse ja töökindluse, seda ka tagasiulatuvalt. Kõik vead, mis on kunagi parandatud, peaksid ka edaspidi jääma parandatuks ja olema kindlustatud automaattestidega. Kõik uued arendused peaksid olema käsitsi üle testitud ja vajadusel ka automaattestidega kaetud.

Arendustööde teostamisel rakendati neljanädalasi sprinte. Iga neljanädalase sprindi hinnanguline maht jäi vahemikku 20 – 30 töötundi.

8.3 Arendustöö käik

Arendustöö oli kõrgelt vaadates jagatud viieks iteratsioonideks.

Esimeses iteratsioonis keskenduti arenduskeskkonna seadistamisele ja tagarakenduse arendustega alustamisele. GitLabi loodi uus koodihoidla. Visual Studios loodi uus .NET Core Web API serverirakendus. Teostati rakenduse seadistamine ja vajalike väliste teekide lisamine. Andmemudeli põhjal loodi domeeniklassid. Domeeniklasside põhjal loodi olemite konfiguratsioonid. Olemite konfiguratsioonide põhjal loodi andmebaas. Iteratsiooni lõpuks oli arenduskeskkond seadistatud ning tagarakenduse struktuur paigas. Lisaks oli valminud ka lokaalne andmebaas koos testandmetega.

Teises iteratsioonis keskenduti tagarakenduse funktsionaalsustele. Rakenduse ärikihti loodi vastavad äriteenused, mis oskavad andmebaasiga suhelda. Veebiliidesesse loodi kontrollerid ja veebiteenused. Iteratsiooni lõpuks oli võimalik läbi veebiteenuste teostada ärioloogiliselt lihtsamaid toiminguid ja andmeoperatsioone.

Kolmandas iteratsioonis keskenduti tagarakenduse keerulisematele toimingutele ja andmeoperatsioonidele. Iteratsiooni lõpuks olid kõik ettenähtud funktsionaalsused tagarakenduse tasemel teostatud ja rakendus oskas saata e-kirju.

Neljandas iteratsioonis keskenduti eesrakenduse loomisele. Loodi uus Angulari klientrakendus. Eesrakenduse funktsionaalsused grupeeriti moodulite järgi ja seejärel loodi moodulites sisse funktsionaalsustele vastavad komponendid ja kasutajaliidese vaated. Vaated ühendati teenustega ja teenused omakorda tagarakenduse veebikihiga. Erinevate vaadete vahele loodi navigeerimise võimalus. Iteratsiooni lõpuks oli tagarakendus lõppkasutaja jaoks kasutatav, aga mitte veel viimistletud kujul.

Viiendas iteratsioonis keskenduti kasutajate autentimisele ja autoriseerimisele ning eesrakenduse viimistlemisele. Tagarakendusele tutvustati kasutaja olemit ja loodi kasutajate autentimiseks vajalikud veebiteenused. Eesrakendusse loodi sisselogimise vaade ja kasutajasessioonide haldus. Viimasena teostati eesrakendusele visuaalne viimistlemine. Iteratsiooni lõpuks sai kasutaja rakendusse sisse logida ja kõiki ettenähtud funktsionaalsusi kasutada.

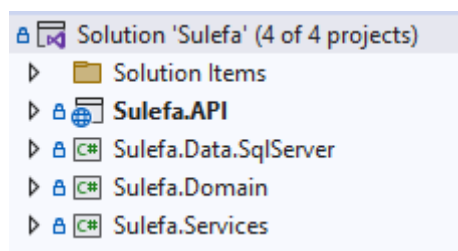
8.4 Tagarakendus

Järgnevalt tutvustatakse arendustööde käigus loodud tagarakendust, mis käitub loodava lahenduse serveripoolse osana. Tagarakenduse peamiseks ülesandeks on olla vahelüli välismaailma ja andmebaasi vahel, teenindades ja töödeldes veebiliidesesse jõudvaid päringuid.

Tagarakenduse loomisel on kasutatud Visual Studio ASP.NET Core Web API projekti põhja. Kasutusel on .NET-i versiooni .NET 6.0 (Long Term Support), millele Microsoft pakub vähemalt 3-aastast tuge.

8.4.1 Tagarakenduse ülesehitus

Tagarakenduse nimi on *Sulefa*. Tagarakendus koosneb ühest .NET rakenduse ülemprojektist (*solution*), mis on omakorda jagatud neljaks alamprojektiks. Alamprojektide ühtse nimetamise huvides kasutatakse kõigi alamprojektide nimedes eesliidet „*Sulefa*.“. Joonisel 1 on kujutatud tagarakenduse projektide struktuur Visual Studio kataloogipuus.



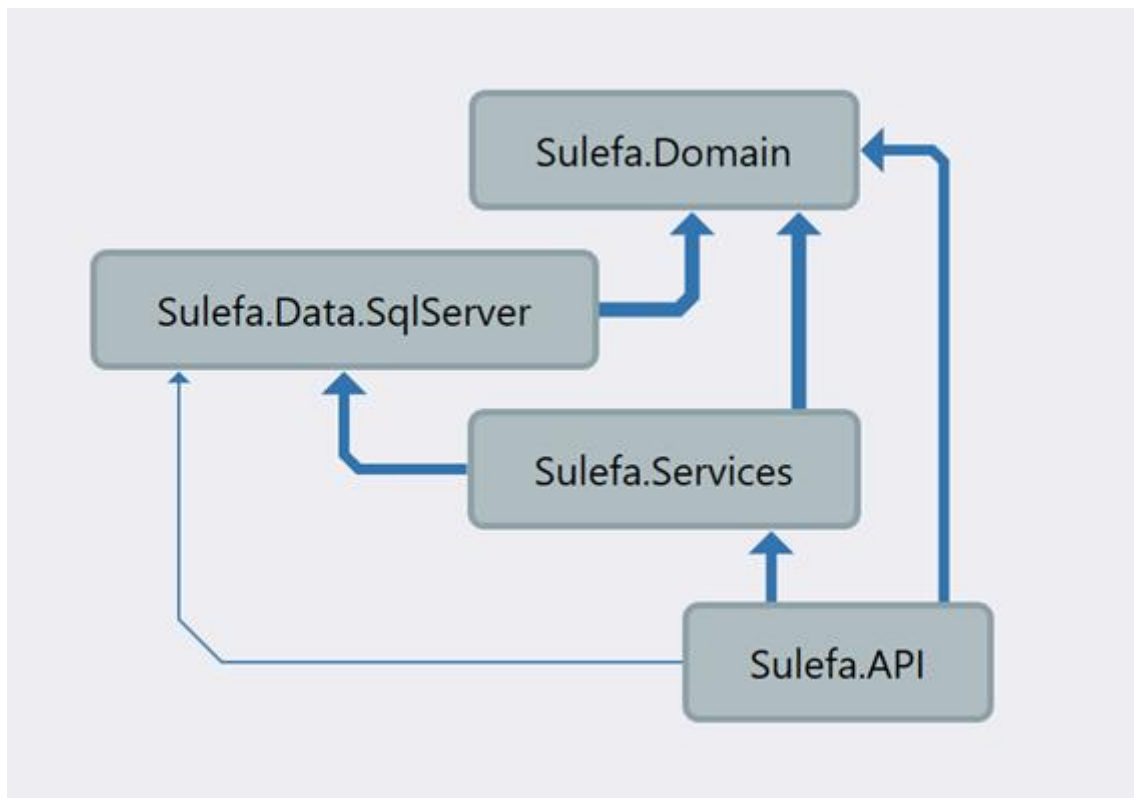
Joonis 1. Tagarakenduse projektide struktuur Visual Studio kataloogipuus.

Tagarakendus koosneb järgnevatest projektidest:

- Veebiliides (*Sulefa.API*)
- Äriteenused (*Sulefa.Services*)
- Äriobjektid (*Sulefa.Domain*)
- Andmebaasi seadistused (*Sulefa.Data.SqlServer*)

Veebiliidese projekti eesmärk on võimaldada välistel klientrakendustel suhelda rakenduse teenuskihiga. Äriteenuste projekti eesmärk on võimaldada rakendusel suhelda andmebaasiga. Äriobjektide projekti eesmärk on hoiustada domeeni klasse, mis on üle tagarakenduse ühises kasutuses. Andmebaasi seadistuste projekti eesmärk on hoolitseda selle eest, et andmebaasi kuju oleks vastavuses rakendusega.

Tulenevalt tagarakenduse mitmekihilisest arhitektuurist on projektide vahel teatud rakendusesisesed sõltuvused. Joonisel 2 on kujutatud tagarakenduse projektide omavahelised rakendusesisesed sõltuvused.



Joonis 2. Tagarakenduse projektide omavahelised rakendusesisesed sõltuvused.

8.4.2 Veebiliides

Tagarakenduse veebiliides asub projektis *Sulefa.API*.

Veebiliides koosneb peamiselt veebiteenustest. Veebiteenused on omakorda jagatud kontrolleriteks. Iga kontrolleri päises on märgitud konkreetse kontrolleri marsruut. Kontrollerite nimed ja nende marsruudid (*routes*) on nimetatud nimetava käände mitmuses [25].

Tagarakenduse kontrollerid pakuvad *RESTful*-tüüpi veebiteenuseid. REST on kaasaegse veebiarenduse kõige populaarsem API arendustehnoloogia [26]. Veebiliidese vaikimisi sõnumite formaadiks on JSON (*JavaScript Object Notation*). JSON on masin-masin veebiliideste puhul maailmas laialdaselt kasutatav andmevorming [27].

GET-tüüpi veebiteenused on andmekirjete lugemiseks. POST-tüüpi veebiteenused on uute andmekirjete lisamiseks ja ühekordsete toimingute sooritamiseks. PUT-tüüpi veebiteenused on olemasolevate andmekirjete uuendamiseks. DELETE-tüüpi veebiteenused on olemasolevate andmekirjete kustutamiseks.

Vaikimisi nõuvad peaaegu kõik kontrollerid ja veebiteenused, et päringut sooritav kasutaja oleks ennast autentitud. Veebiliides peab autentitud kasutajate üle ise arvet. Kasutaja autentimist ei nõuta ainult üksikutelt veebiteenustelt.

Rakendus on seadistatud nii, et iga sissetuleva HTTP (*HyperText Transfer Protocol*) päringu jaoks alustatakse uut andmebaasi transaktsiooni. Andmebaasi transaktsioon kestab üldjuhul seni, kuni ärikihis toimub andmete salvestamise väljakutse või kui päringu töötlemise käigus tekib mõni ootamatu viga. Selline käsitlus tagab päringute töötlemisel transaktsioonilise terviklikkuse, kuna see vähendab riski, et andmebaas võiks poolikute andmeoperatsioonide tõttu sattuda vigasesse seisusse.

Kui serverisse jõudnud HTTP päring on autentne ja leidnud tee õige kontrolleriini ning sealt edasi ka õige veebiteenuseni, siis algab päringu äriloogiline töötlemine. Selleks pöördub veebiteenus ise vastavate äriteenuste poole.

8.4.3 Äriteenused

Tagarakenduse äriteenused asuvad projektis *Sulefa.Services*.

Ärikihis paikneb enamik tagarakenduse äriloogikast. Äriteenused töötlevad veebiteenustes tehtud pöördumisi ja vastavad neile.

Äriteenused on grupeeritud olemite kaupa. Kuna käesoleva lõputöö eesmärgiks on luua tulevase rakenduse prototüüp, siis on äriteenused loodud ainult kõige tähtsamatele tagarakenduse olemitele.

Äriteenused suhtlevad andmebaasiga läbi Entity Frameworki (EF). EF on avatud lähtekoodiga ORM-i (*Object-Relational Mapping*) raamistik .NET-i rakendustele. See võimaldab töötada andmetega kõrgemal abstraktsioonitasemel, kasutades domeenispetsiifiliste klasside objekte, seejuures keskendumata andmebaasi aluseks olevatele tabelitele ja veergudele, kus neid andmeid päriselt hoitakse [28].

Äriteenused kasutavad andmebaasipäringute koostamiseks Microsofti eriotstarbelist päringukeelt LINQ (*Language Integrated Query*). LINQ võimaldab andmeallikaga suhtlemisel kasutada objektorienteeritud lähenemist ning päringu koostaja ei pea täpselt teadma, mis kujul andmeallikas andmeid sisemiselt hoiab, sest programm teisendab LINQ päringud ise korrektseteks andmebaasipäringuteks [29]. Süsteemi poolt koostatud SQL päringud ei pruugi küll alati olla sama optimaalsed kui käsitsi kirjutatud SQL päringud, aga sellest eest on LINQ päringud jällegi tugevalt tüübitud (*strongly typed*) ja seetõttu tulevad päringu koostamisel tehtud vead enamasti välja juba rakenduse kompileerimise ajal.

Andmebaasist andmete pärimisel kasutakse võimalusel alati projektsioone. Projektsioon on toiming, mille käigus viiakse esialgne andmeobjekt uuele kujule. Selle asemel, et iga olemit kohta andmebaasist pärida alati kõik andmeväljad, on efektiivsem pärida ainult need andmeväljad, mida läheb programmi koodis hiljem ka päriselt vaja [30]. Projektsioonide kasutamine vähendab märkimisväärselt andmebaasist rakendusele võrgu kaudu edastatavate andmete hulka ja muudab päringud kiiremaks [31].

Andmebaasi andmete lisamisel ja uuendamisel kasutab tagarakendus veebiteenuste ja äriteenuste vahelises suhtluses käsuklasse. Käsuklass on tark andmeedastusobjekt ehk DTO (*Data Transfer Object*), mis hoiab endas metainformatsiooni lisatava või muudetava andmekirje kohta. Kõigi käsuklasside nimetused on järelliitega “*Cmd*” ehk “*Command*”. Üldiselt sisaldavad kõik käsuklassid andmeid loodava või lisatava

andmekirje kohta, aga võivad sisaldada ka muud metainformatsiooni ja ärioloogikat, mida andmekirje loomise või lisamise käigus võib vaja minna.

Andmekirjete loomiseks mõeldud käsuklassid algavad algava eesliitega “*Create*”. Neid kasutatakse veebiliidese ja äriteenuste vahelises suhtluses uute andmekirjete lisamiseks. Lisaks oskavad antud käsuklassid ka iseennast olemiteks teisendada.

Andmekirjete muutmiseks mõeldud käsuklassid algavad eesliitega “*Update*”. Neid kasutatakse veebiliidese ja äriteenuste vahelises suhtluses olemasolevate andmekirjete uuendamiseks.

8.4.4 Andmebaas

Tagarakendus kasutab andmebaasina Microsoft SQL Serverit. Andmebaasi seadistused asuvad projektis *Sulefa.Data.SqlServer*.

Rakendusele andmebaasi loomiseks ja jooksvaks uuendamiseks kasutatakse lähenemist *code-first*. *Code-first* puhul keskendutakse rakenduse äridomeenile ja rakenduse loomist alustatakse äriolemeid kujutavate klasside loomisest, mille põhjal genereeritakse valmis andmebaas ja sinna sisse klassidele vastavad andmetabelid. *Code-first* lähenemine on peamiselt kasutuses domeenipõhises disainis (*Domain-Driven Design* ehk DDD) [32].

Käesolevas projektis on domeeni klassid eraldatud enda andmebaasi kujust, sest domeeni klassid ei pea olema teadlikud, kus või millisel kujul neid andmebaasis kujutatakse. Selline lähenemine võimaldab tagarakenduses kasutatavad olemid siduda lahti andmebaasis asuvatest andmetabelitest. Seeläbi on rakendus tulevikukindlam, sest rakenduse kood ei sõltu nii otseselt konkreetsel juhul kasutusel olevast andmebaasimootorist ega viisist, kuidas andmebaas andmeid sisemiselt hoiustab. Domeeni klassid asuvad projektis *Sulefa.Domain* ja olemite konfiguratsioonid asuvad projektis *Sulefa.Data.SqlServer*.

Code-first lähenemise puhul teostatakse andmebaasi muudatusi migratsioonide abil. Migratsioonid on klassid, mis kontrollivad andmebaasile rakendatavaid muudatusi. Migratsioonid võivad sisaldada nii andmebaasi skeemi muudatusi kui ka sisulisi muudatusi andmetabelite andmetes. Migratsioonid kuuluvad samuti tagarakenduse koodi hulka.

Andmebaasis on kõikide andmetabelite nimed ainsuse nimetavas käändes. Andmetabelite nimetamisel ollakse seisukohal, et ainsuse vormi kasutamine on kõige sobilikum, kuna üksikule andmebaasikirjele viidates oleks segadust tekitav kasutada mitmuse nimetavat vormi [33].

Rakenduse andmemudel on normaliseeritud ja andmetabelid on viidud kolmandale normaalkujule (3FN ehk *third normal form*). Normaliseerimine põhineb kindlate, kuid samas suhteliselt üldise iseloomuga, piirangute kogumitel, mida rakendatakse andmemudelile [34]. Normaliseerimise abil on võimalik vähendada andmete kordumist ehk liiasust ja sellest tingitud omavahelisi vastuolusid [35].

Olemite omavahelised põlvnemissuhted on andmemudelis lahendatud kompositsiooni abil. Näiteks on kõik järgnevad olemid kaudselt tegelikult isikud: kandidaat (*Applicant*), hädaabikontakt (*EmergencyContact*) ja ülalpeetav (*Dependant*). Seetõttu peaksid neil kõigil olema ka isiku (*Person*) tunnused. Lisaks on hädaabikontakt ja ülalpeetav samal ajal ka veel pereliikmed (*FamilyMember*). Kõikide isikutunnuseid omavate olemite andmetabelite külge isikuga seotud veergude lisamine läheks aga vastuollu DRY (*Do Not Repeat Yourself*) printsiibiga, mille kohaselt tuleks rakenduste arendamisel dubleerimist vältida ja püüda sarnase tähendusega andmeid hoida ühesainsas kohas [36]. Seetõttu on mõistlik isikutunnuseid hoida ainult isiku tabelis. Kõik teised andmetabelid, mis vajavad isikutunnuseid, saavad ennast isiku tabeli abil laiendada.

Andmebaasis on kasutusel ka mõned sildtabelid (*bridge tables*). Sildtabelid võimaldavad erinevaid andmetabeleid omavahel kokku siduda, defineerides nende vahele mitumitmele seose [37]. Sildtabelite kasutamine hõlbustab sama tähendusega andmete paigutamist samasse kohta ja vähendab samalaadsete andmeveergude dubleerimist üle andmebaasi. Võtame näiteks olemid kandidaat (*Applicant*) ja hädaabikontakt (*EmergencyContact*). Neil mõlemal olemil võib olla kontaktandmeid, mida on vaja andmebaasi salvestada. Kõiki kontakte oleks aga mõistlik hoida andmebaasis ühes ja samas tabelis (*Contact*). Kontaktandmete tabel aga ei pea temast sõltuvuses olevatest tabelites teadma mitte midagi. Kõigi kolme tingimuse rahuldamiseks oleks vaja tabelite kandidaat (*Applicant*) ja kontaktandmete kirje (*Contact*) vahele lisada sildtabel *ApplicantToContact*. Sama on vaja teha ka hädaabikontaktiga, lisades tabelite hädaabikontakt (*EmergencyContact*) ja kontakt (*Contact*) vahele sildtabeli *EmergencyContactToContact*. Selliselt toimides on võimalik kandidaadi ja

hüdaabikontakti külge märkida kontaktandmeid nii, et kontaktandmete tabel (*Contact*) ise ei tea kummaski seoses olevast andmetabelist mitte midagi.

Normaliseerimise, kompositsiooni ja sildtabelite abil on võimalik disanida paindlikke andmebaasiskeeme, mis on võimalised kohanema vastavalt ärivajadustele. Näiteks kui tulevikus peaks ilmneva vajadus reaalandmetele lähedaste andmetega rakendust testida, siis on andmebaasi obfuskeerimine lihtsam, sest kõik tundliku sisuga andmed asuvad kindlaksmääratud kohtades ega ole üle andmebaasi laiiali.

8.5 Eesrakendus

Järgnevalt tutvustatakse töö käigus loodud eesrakendust, mis käitub loodava lahenduse kliendipoolse osana. Eesrakenduse ehk klientrakenduse peamiseks ülesandeks on olla vahelüli kasutaja ja serveris asuva tagarakenduse vahel, võimaldades kasutajal suhelda tagarakendusega läbi interaktiivse graafilise kasutajaliidese [38].

Eesrakendus on loodud kasutades Angulari veebiraamistikku. Kasutusel on kõige uuem Angulari versioon, Angular 15, millele pakutakse aktiivset tuge veel kuni 2024. aasta mai keskpaigani. Klientrakenduse skelett on loodud Angular CLI (*Command Line Interface*) abil. Angular CLI on mugav viis uute Angulari rakenduste loomiseks ja olemasolevate arendamiseks ning hooldamiseks [39].

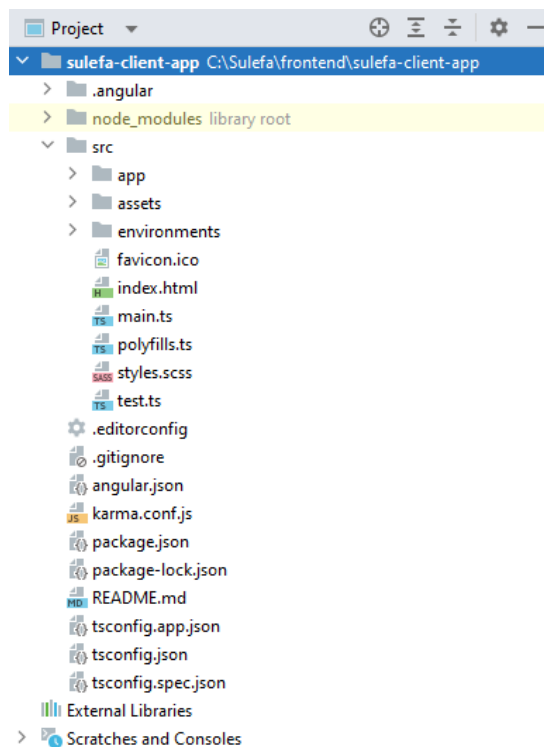
Loodud klientrakendus on üheleherakendus ehk SPA (*Single Page App*). Üheleherakendused on hetkel kõige populaarsem viis veebirakenduste loomiseks, sest need laadivad väga kiiresti. Kuna vaadete vahel navigeerimine ja suur osa arvutustest ning toimingutest tehakse ära juba kasutaja brauseris ilma vahepeal veebiserveri poole pöördumata, siis seetõttu on ka üheleherakenduste üleüldine kasutuskogemus parem [40].

Eesrakenduse kasutajaliides on adaptiivne ja rakendus on kasutatav nii töölaua- kui ka mobiilivaates. See tagab sujuva ja ühtlase kasutajakogemuse ükskõik millise seadme või brauseriga.

8.5.1 Eesrakenduse ülesehitus

Eesrakenduse nimi on *sulefa-client-app*. Angulari veebirakendusele omaselt koosneb see moodulitest, komponentidest, vaadetest, teenustest, stiilifailidest, mudelitest ja seadistuse

failidest. Joonisel 3 on kujutatud eesrakenduse projekti struktuur WebStormi kataloogipuu.



Joonis 3. Eesrakenduse projekti struktuur WebStormi kataloogipuu.

Rakenduse arhitektuur põhineb Angulari meeskonna poolt välja töötatud kontseptsioonidel ja parimatel praktikatel. Rakenduse põhilisteks ehitusklotsideks on komponendid. Komponendid esindavad kasutajaliidese mingit osa ja on üldjuhul korduvkasutatavad ning võivad omakorda koosneda muudest komponentidest [41]. Komponente struktureeritakse funktsionaalselt ja ärioloogiliselt terviklikeks komplektideks – mooduliteks.

Moodulid on Angulari raamistiku poolt ettenähtud viis rühmitada rakenduses kasutatavaid komponente nii, et neid saaks omavahel kombineerida ka teiste moodulitega [42]. Rakenduse mooduleid on võimalik panna laisalt laadima (*lazy loading*). Laisk laadimine võimaldab mooduleid laadida alles siis, kui nendes peituvaid komponente ka päriselt vaja läheb. See parandab olulisel määral rakenduse jõudlust ja vähendab märkimisväärselt esimese vaate laadimisaega [43].

Komponendid kasutavad teenuseid. Teenused on mitmetähenduslikud objektid, pakkudes komponentidele mis tahes väärtusi või ärioloogilisi funktsioone, mida rakendus toimimiseks vajab. Tavaliselt on teenused siiski kitsa ja täpselt määratletud eesmärgiga

klassid. Teenused sobivad hästi näiteks ülesannete jaoks nagu andmesuhtlus serveriga ja kasutaja sisendi valideerimine. Angular eristab komponente teenustest, et suurendada rakenduse modulaarsust ja korduvkasutatavust [44].

Selleks, et kasutaja saaks eesrakenduse kasutajaliideses erinevate vaadete vahel ringi liikuda, kasutab eesrakendus marsruutimist (*routing*) [45]. Marsruutimist pakub Angulari raamistikku sisseehitatud klass *RouterModule* ehk marsruuter (*router*). Marsruuteril on Angulari rakendustes oluline roll, kuna selle abil kaardistatakse ära kõikide veebiressursside URL-id (*Uniform Resource Locator*) ja seatakse need vastavusse konkreetsete komponentidega. Selline lähenemine võimaldab kasutajatel linkide ja brauseri edasi-tagasi nuppude abil rakenduse sees sujuval ja arusaadaval viisil ringi liikuda.

8.5.2 Suhtlus tagarakendusega

Eesrakendus suhtleb tagarakendusega läbi *RESTful* veebiteenuste. Kogu suhtlus käib üle hüpertexti edastusprotokolliga HTTP. Selleks, et eesrakendus saaks tagarakendusega suhelda, on kasutusel Angulari raamistikku sisseehitatud klass *HttpClient*. *HttpClient* võimaldab klientrakendusel sooritada HTTP päringuid ja saada tagasi vastuseid ning neid vastuseid vastavalt vajadusele ka töödelda.

Enamik tagarakenduse veebiteenuseid teenindavad ainult autentitud kasutajate päringuid. Autentimise tõestamiseks paneb eesrakendus iga tagarakenduse pihta saadetava päringu päisesse kaasa päringut sooritava kasutaja kehtiva juurdepääsuloa. Väljuvate päringute päiste korrektse andmekoosseisu eest hoolitseb päringute pealtkuulaja ehk *interceptor*. Päringute pealtkuulaja on vahevara (*middleware*) klass, mis oskab kinni püüda väljuvaid päringuid ja saabuvaid vastuseid, neid vajadusel modifitseerida ja lõpuks neid teatepulgana rakenduse järgmistele töötlejatele edasi anda [46].

8.5.3 Kujundus ja stiilid

Rakenduse kasutajaliidese kujundamiseks on kasutatud Angular Materiali komponente.

Angular Material on Angulari ametlik kujundusraamistik ja kasutajaliidese komponentide teek. Angular Materiali komponendid aitavad luua atraktiivseid, funktsionaalseid ja ühtlase väljanägemisega veebirakendusi, järgides samal ajal kaasaegse veebidisaini põhimõtteid. Angular Material on inspireeritud Google'i disainsüsteemist nimega

Material Design [47]. Material Design on vastavalt vajadusele kohandatav juhiste, komponentide ja tööriistade süsteem, mis toetab ja propageerib kasutajaliidese kujundamisel üldtunnustatud praktikate kasutamist [48].

Kasutajaliidese vaateid kujundatakse HTML-i (*HyperText Markup Language*) ja SASS-i (*Syntactically Awesome Style Sheets*) abil. SASS on süntaksi täiustustega CSS-i (*Cascading Style Sheets*) eelprotsessor ja skriptimiskeel. SASS-i süntaksis kirjutatud stiililehed teisendatakse programmi poolt automaatselt ümber tavapärasteks, veebibrauseritele arusaadavateks CSS-i stiililehtedeks. SASS võimaldab rakenduse stiililehtedes kasutada pärilust ja muutujaid [49]. Tänu sellele on rakenduses kokkuvõttes vähem stiilidega seotud koodi ja loodetavasti on selline kood ka paremini organiseeritud [50].

8.6 Turvalisus

Kasutajate autentimist teostatakse JWT (*JSON Web Token*) juurdepääsulubadega. Eduka autentimise korral tagastab veebiliides eesrakendusele lühiajalise JWT juurdepääsuloa, mis aegub 15 minuti pärast ja värskendusloa, mis aegub 7 päeva pärast. Juurdepääsuluba kasutatakse veebiliideses turvatud veebiteenustele ligipääsemiseks. Värskendusloa kasutatakse uute juurdepääsulubade genereerimiseks. Turvalisuse suurendamiseks talletatakse värskenduslubasid HTTP küpsistes, kuna need ei ole juurdepääsetavad kliendipoolsele JavaScriptile, mis vähendab riski korraldada XSS (*Cross-Site Scripting*) rünnakuid. Lisaks on värskenduslubadel veebiliideses juurdepääs ainult uute juurdepääsulubade loomiseks, mis takistab nende kasutamist CSRF-i (*Cross-Site Request Forgery*) rünnakutes. Kui kasutaja rakendusest välja logib tühistatakse ka kõik talle väljastatud värskendusload [51].

Iga autenditud kasutaja näeb ainult enda asutuse andmeid. Teiste asutuste andmete nägemine on andmebaasi päringute tasemel piiratud, sest igale päringule pannakse kaasa päringut sooritava kasutaja asutuse identifikaator.

Selleks, et kandidaat saaks tutvuda talle saadetud personaalse tööpakkumisega ja täita uue töötaja isikuandmete ankeeti, antakse talle ajutine piiratud õigustega ligipääs rakendusele.

Süsteem loob personaalse tööpakkumise jaoks unikaalse avaliku lingi ning see link saadetakse kandidaadile meili peale. Kõnealuse lingi avamine viib lingi avaja personaalse

tööpakkumise vaatesse. Kui antud tööpakkumine on juba kinnitatud, siis selle kohta tagarakendus enam andmeid ei väljastata ja kasutaja suunatakse kinnitatud tööpakkumise vaatesse. Seal antakse kasutajale võimalus suunduda isikuandmete ankeeti täitma. Isikuandmete ankeedi täitmise võimalust pakutakse ainult siis, kui see on antud juhul ettenähtud ja ankeet on veel täitmata. Kinnitatud tööpakkumise vaates on kandidaadil võimalik ka tööpakkumisele antud kinnitus tühistada. Tööpakkumise kinnitamise tühistamise võimalust pakutakse kandidaadile ainult siis, kui andmed on veel tööandja poolel menetlemata. Tööpakkumise tühistamise käigus kustutatakse süsteemist ära ka kõik kandidaadi poolt esitatud andmed. Kui tööpakkumine, mida avatakse on vanem kui 30 päeva, siis antakse kasutajale tööpakkumise olemasolu kohta umbmäärane vastus, et mitte paljastada andmete eksisteerimist, sest tegemist võib olla andmete õngitsemisega.

Isikuandmete ankeet on samuti personaalne ja unikaalse avaliku lingiga. Ankeet võib esmakordsel avamisel sisaldada vähesel määral kandidaadi isikuandmeid, näiteks eesnimi, perekonnanimi ja meiliaadress. Kui isikuandmete ankeeti avatakse mitmendat korda ja see on mõnel eelmisel korral juba ära täidetud ning esitatud, siis selle kohta tagarakenduses enam andmeid ei väljastata ja kasutaja suunatakse esitatud ankeedi vaatesse. Esitatud ankeedi vaates on kandidaadil samuti võimalik tööpakkumisele antud kinnitus tühistada. Tööpakkumise kinnitamise tühistamise võimalust pakutakse kandidaadile ainult siis, kui andmed on veel tööandja poolel menetlemata. Kui kasutaja soovib enda esitatud andmeid kontrollida, siis saab ta esitatud ankeedi vaates tellida uue ajutise ligipääsu täidetud ankeedile. Seejärel saadab süsteem kasutaja meili peale uue unikaalse lingi ja isikuandmete edastamise ring algab otsast peale. Lingi abil on kandidaadil võimalik esitatud andmetega tutvuda ja neid muuta. Kui isikuandmete ankeet, mida avatakse on seotud tööpakkumisega, mis on vanem kui 30 päeva, siis antakse kasutajale ankeedi olemasolu kohta umbmäärane vastus, et mitte paljastada andmete eksisteerimist, sest tegemist võib olla andmete õngitsemisega.

8.7 Testimine

Tarkvara testimine on arendusprotsessi lahutamatu osa, sest sarnaselt muule toodangule, millega ettevõtted tegelevad, peab ka tarkvara pakkumisel veenduma, et selles poleks vigu ega tehnilist praaki [52].

Testimise käigus hinnatakse ja kontrollitakse, kas tarkvara teeb seda, mida ta peaks tegema. Pakkudes kliendile koheselt kõrge kvaliteediga tarkvara on teenusepakkujal kaudselt võimalik vähendada arenduskulusid ja maandada riske. Testimise abil on võimalik vigu varakult avastada või neid üldsegi vältida. Varajases arendusfaasis avastatud puuduste kõrvaldamine on hinnanguliselt kordades odavam kui hiljem juba toodangukeskkonnas ilmnunud vigadega tegelemine [53]. Hilinenud tarne ja tarkvara kirjutamisel tehtud vead võivad pöördumatult kahjustada teenusepakkuja mainet ning rikkuda häid kliendisuhteid [54].

Rakenduse testimisel kasutati nii käsitsi testimist kui ka automaattestimist. Rakenduse käsitsi testimine tuvastab tavaliselt enamiku probleemidest, sealhulgas ka puudused rakenduse välimuse ja kasutusmugavuse osas. Paraku on aga käsitsi testimine aeglane ja ebaefektiivne, kuna testija peab mitmeid kordi tegema läbi täpselt samu toiminguid ning tihti võib testi läbiviimiseks vajalike andmete ettevalmistamine võtta isegi kauem aega kui testimine ise [55]. Automaattestimise läbiviimine on seevastu aga kiire ja saadav tagasiside vahetu ning alati täpne. Automaatteste võib reeglina käivitada ükskõik mis hetkel ja kui testid on juba kord valmis kirjutatud, siis võtab ka keeruliste stsenaariumite ettevalmistamine ja läbi testimine vaid mõne hetke.

Testimisel lähtuti peamiselt eelmises peatükis kirjeldatud funktsionaalsetest ja mittefunktsionaalsetest nõuetest. Lisaks vaadeldi ka üleüldist kasutajakogemust ja rakenduse toimimist.

Tagarakenduse veebiteenuste käsitsi testimiseks kasutati tööriistaid Swagger ja Postman. Swagger on *OpenAPI* spetsifikatsioonil põhinev avatud lähtekoodiga tööriistade komplekt, mis lihtsustab *RESTful*-tüüpi veebiteenuste kujundamist, ehitamist, dokumenteerimist ja tarbimist [56]. Postman on veebiteenuste dokumenteerimise ja testimise platvorm, mis lihtsustab ja automatiseerib olulisel määral veebiteenuste tarbimist [57].

Tagarakenduse automaattestimiseks kasutatakse xUnit-i raamistikus loodud ühikteste ja integratsiooniteste. xUnit on tasuta avatud lähtekoodiga kogukonnakeskne automaattestimise tööriist, mis on mõeldud .NET raamistikul loodud rakenduste testimiseks [58]. Ühiktestide abil testiti peamiselt tagarakenduse ärioloogikakihti, andmeobjektide transformeerimist ühelt kujult teisele ja korrektsete veateadete

väljastamist. Integratsioonitestide abil testiti tagarakenduse veebiliidest ja päringute ning käskude täitmise transaktsioonilist terviklikkust.

Andmebaasis tehti kontrollpäringuid SQL Server Management Studio (SSMS) abil. Kontrollpäringud olid vajalikud veendumaks, et tagarakendus on andmemudeli korrektselt üles ehitanud ja et peale teatud toiminguid oleksid andmed ootuspärasest seisundis.

Eesrakendust testiti käsitsi, mängides läbi erinevaid stsenaariume ja levinud kasutusjuhtusid, millega programmi kasutajad kindlasti igapäevaselt kokku puutuvad.

Rakenduse kaudu meilide saatmise jaoks kasutati veebirakendust Ethereum. Ethereum on tasuta SMTP-teenus (*Simple Mail Transfer Protocol*), mille abil on võimalik mängida läbi meilide saatmine ja vastuvõtmine, kusjuures päris sõnumeid ei edastata protsessi käigus kunagi [59]. Saadetud ja vastuvõetud meilidega saab tutvuda Ethereumi veebikeskkonnas, kuhu peab selle jaoks eelnevalt kasutajakonto looma.

9 Loodud lahendus

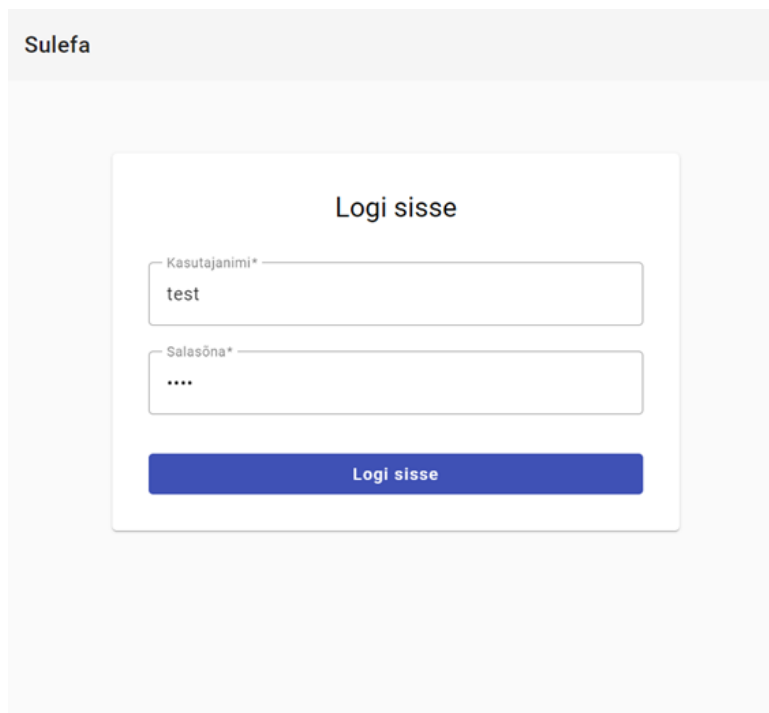
Järgnevas peatükis tutvustatakse loodud lahendust ja analüüsitakse saavutatud kasutatavust. Lisaks viiakse läbi mõned kasutajatestid ja võrreldakse lahenduse vastavust lähtetingimustele. Viimasena analüüsitakse võimalikku majanduslikku kasu ja kaardistatakse võimalusi edasiarendusteks.

9.1 Lahenduse kasutamine

Alljärgnevalt antakse ülevaate rakenduse kasutamisest lõppkasutaja pilgu läbi.

9.1.1 Sisselogimine

Personalitöötajal on vaja rakenduse kasutamiseks sisse logida (Joonis 4). Ilma autentimata pole personalitöötajal võimalik rakendust kasutada. Vaikimisi kestab kasutajasesioon 15 minutit. Kui kasutaja vahepeal välja ei logi, siis enne kasutajasesiooni lõppu küsib eesrakendus serverist ise uue identsustõendi ning sujuvalt algab uus kasutajasesioon.

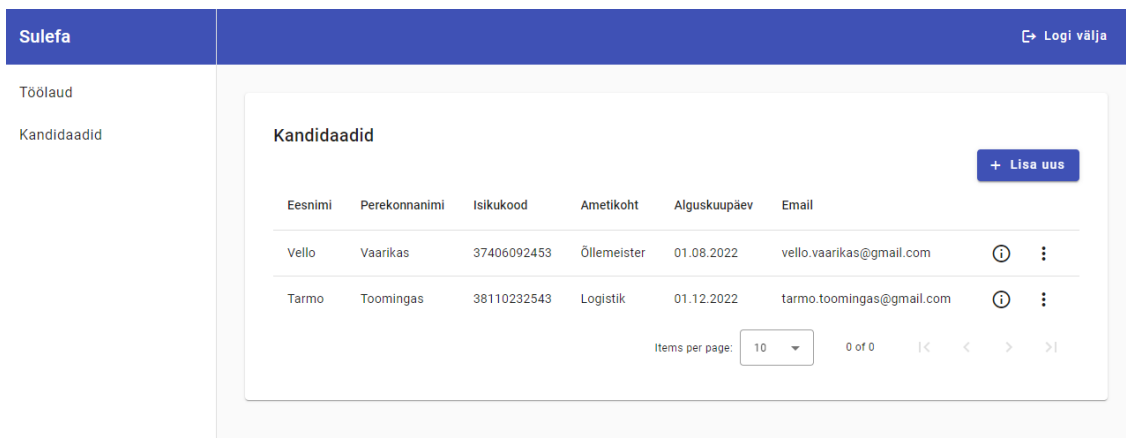


Screenshot of the login page (Joonis 4). The page is titled "Sulefa" and contains a "Logi sisse" (Login) form. The form has two input fields: "Kasutajanimi*" (Username) with the value "test" and "Salasõna*" (Password) with masked characters "....". Below the fields is a blue button labeled "Logi sisse".

Joonis 4. Sisselogimise vaade.

9.1.2 Kandidaatide haldamine

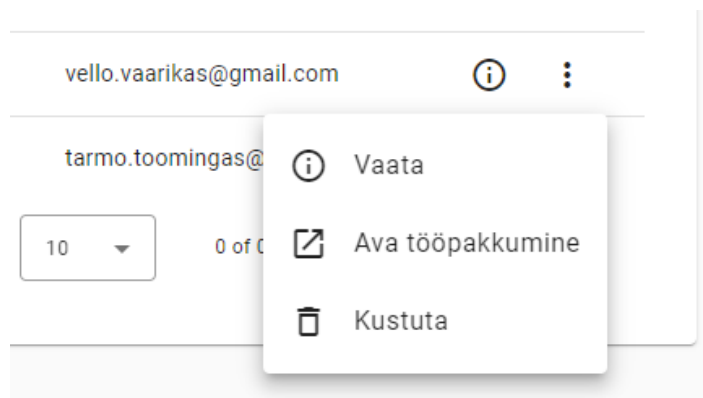
Kui personalitöötaja rakendusse sisse logib, suunatakse ta automaatselt kandidaatide nimekirja vaatesse (Joonis 5). Seal saab personalitöötaja kandidaate hallata.



Eesnimi	Perekonnanimi	Isikukood	Ametikoht	Alguskuupäev	Email		
Vello	Vaarikas	37406092453	Öllemeister	01.08.2022	vello.vaarikas@gmail.com	ⓘ	⋮
Tarmo	Toomingas	38110232543	Logistik	01.12.2022	tarmo.toomingas@gmail.com	ⓘ	⋮

Joonis 5. Kandidaatide nimekiri.

Iga kandidaadi rea lõpus on tegevuste ja toimingute nupud (Joonis 6). Personalitöötaja saab kandidaate vaadata, muuta, lisada ja kustutada. Üksiku kandidaadi andmete kuvamiseks ja muutmiseks suunatakse kasutaja konkreetse kandidaadi vaatesse. Uue kandidaadi lisamine käib modaalakna kaudu.



Joonis 6. Reasisesed tegevused ja toimingud kandidaatidega.

9.1.3 Kandidaadi lisamine ja tööpakkumise saatmine

Süsteemi uue kandidaadi lisamiseks ja talle personaalse tööpakkumise saatmiseks tuleb kandidaatide vaates vajutada nupule „Lisa uus“. Seejärel avaneb modaalaken pealkirjaga „Lisa uus kandidaat“ (Joonis 7).

Personalitöötaja täidab ära kandidaadi kohta kõik teada olevad isikuandmed ja valib rippmenüüst ametikoha, mida kandidaadile pakutakse. Lisaks valib personalitöötaja ka lepingu liigi ja orienteeruva tööle asumise kuupäeva ning märgib ära töökoormuse.

Eelviimasena täidetakse ära tasustamise info. Kui personalitöötaja soovib tööpakkumise saata kandidaadi meilile koheselt, siis teeb ta selleks vormi lõpus vastava valiku. Vaikimisi on antud valik tegemata ja kandidaadile tööpakkumist automaatselt ei saadeta.

The image shows two side-by-side screenshots of the 'Lisa uus kandidaat' (Add new candidate) form in the Sulefa system. The left screenshot shows the 'Kandideadi andmed' (Candidate's data) section with fields for 'Eesnimi*' (First name) containing 'Mari', 'Perekonnanimi*' (Last name) containing 'Maasikas', and 'E-mail*' (Email) containing 'mari.maasikas@gmail.com'. Below this is the 'Ametikoha andmed' (Job position data) section with 'Ametikoht*' (Job position) set to 'Turundusjuht' and 'Lepingu liik*' (Contract type) set to 'Tööleping'. The right screenshot shows the 'Tasustamine' (Salary) section with 'Töötasu liik*' (Salary type) set to 'Kuupalk' and 'Töötasu (bruto)*' (Gross salary) set to '€ 2750'. It also shows 'Teavitused' (Notifications) with a checked box for 'Saada tööpakkumine kandidaadile meili peale' (Send job offer to candidate by email). Both screenshots have a 'Salvesta' (Save) button at the bottom.

Joonis 7. Uue kandidaadi lisamise modaalaken.

Kui personalitöötaja on sisestatud andmetega rahul, siis vajutab ta nuppu „Salvesta“. Rakendus valideerib salvestatavaid andmeid ja teostab nendega mitmesuguseid kontrolle. Muuhulgas kontrollitakse ka seda, ega selliste andmetega kandidaati pole juba süsteemi varasemalt sisestatud. Vigade ilmnmisel teavitatakse kasutajat vastavasisulise veateatega.

Kui andmed on süsteemi poolt valideeritud ja konflikte ei esine, siis kandidaat salvestatakse ja kasutajale kuvatakse lehe allosas informatiivne teade, et kandidaat on edukalt lisatud. Teates on ka otsetee link „Mine lehele“, mis viib kasutaja vastloodud kandidaadi vaatesse (Joonis 8).

Sulefa						Logi välja	
Töölaud							
Kandidaadid							
Kandidaadid							+ Lisa uus
Eesnimi	Perekonnanimi	Isikukood	Ametikoht	Alguskuupäev	Email		
Vello	Vaarikas	37406092453	Õllemeister	01.08.2022	vello.vaarikas@gmail.com	ⓘ ⋮	
Tarmo	Toomingas	38110232543	Logistik	01.12.2022	tarmo.toomingas@gmail.com	ⓘ ⋮	
Mari	Maasikas		Turundusjuht	29.05.2023	mari.maasikas@gmail.com	ⓘ ⋮	

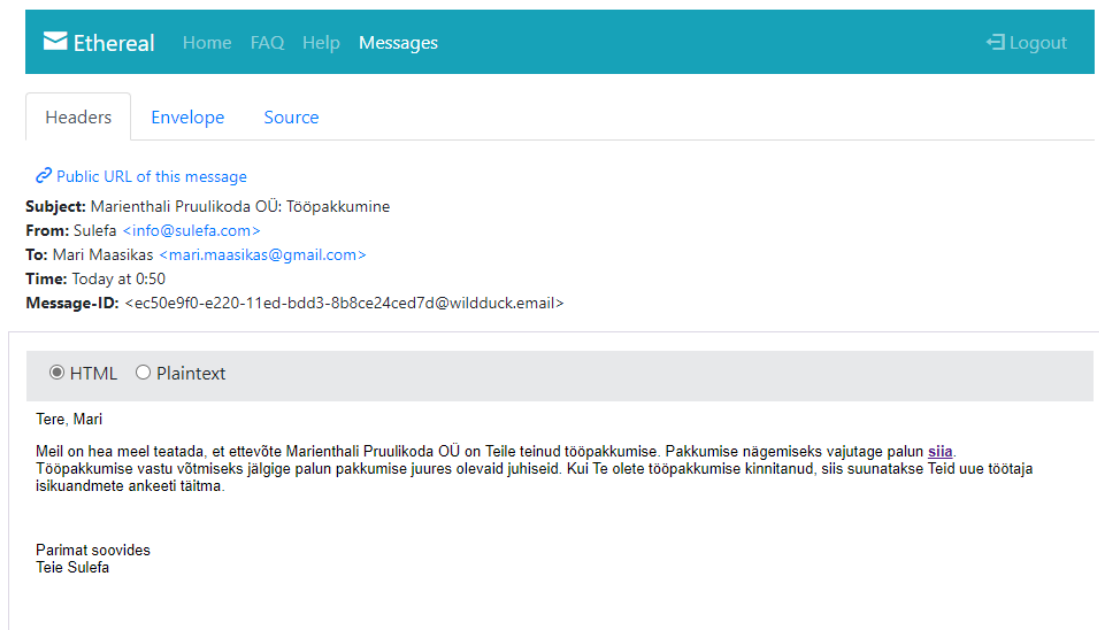
Items per page: 10 0 of 0 < >

Kandidaat 'Mari Maasikas' lisatud **Mine lehele**

Joonis 8. Edukalt lisatud kandidaat.

9.1.4 Tööpakkumise vaatamine ja kinnitamine

Kui personalitöötaja tegi süsteemi uut kandidaati lisades vastava valiku, siis saadetakse koostatud tööpakkumine koheselt kandidaadi meilile (Joonis 9).



Joonis 9. Kandidaadile saadetud tööpakkumise meil.

Kandidaat tutvub talle saadetud elektronkirjaga ja soovi korral vajutab meilis olevale lingile „siia“. Lingile vajutades suunatakse kandidaat personaalse tööpakkumise vaatesse (Joonis 10). Kandidaat saab seal talle tehtud tööpakkumisega tutvuda ja selle kinnitada või tagasi lükata. Kui kandidaat tööpakkumise kinnitab, siis suunatakse ta uue töötaja isikuandmete ankeeti täitma.

Tööpakkumine

Mari Maasikas

Ettevõtte Marienthali Pruulikoda OÜ on Teile teinud tööpakkumise. Pakkumise tingimustega saate tutvuda allpool. Tööpakkumise vastuvõtmiseks vajutage vormi lõpus olevat nuppu 'Kinnita tööpakkumine'. Peale seda suunatakse teid uue töötaja isikuandmete ankeeti täitma.

**Ametikoht:**

Turundusjuht

**Asukoht struktuuris:**

Marienthali Pruulikoda OÜ / Müük / Turundus

**Lepingu tüüp:**

Tööleping

**Tööle asumise kuupäev:**

29.05.2023

**Töökoormus:**

1.0 / 40 tundi nädalas

**Kuupalk (bruto):**

2,750 EUR

**Katseaeg:**

4 kuud

Hüved ja soodustused:

- 7 kalendripäeva lisapuhkust aastas
- Kaugtöö võimalus
- Sporditoetus 33 eurot kuus
- Iga-aastased kogupereüritused ja koosviibimised
- Tasuta parkimine Tallinna kesklinnas



Olen tööpakkumise läbi lugenud ja nõustun selle tingimustega.

Kinnita tööpakkumine

Keeldu tööpakkumisest

Joonis 10. Kandidaadile saadetud tööpakkumine.

9.1.5 Uue töötaja isikuandmete ankeedi täitmine

Peale tööpakkumise kinnitamist suunatakse kandidaat uue töötaja isikuandmete ankeeti täitma (Joonis 11).

Rakendus valideerib ankeedi andmevälju. Kui ankeedi vahekaardil on kohustuslikke andmevälju, mis on täitmata või on täidetud vigaselt, siis kandidaadil pole võimalik ankeedi järgmisele vahekaardile edasi liikuda.

Isikukoodi sisestamisel täidetakse sugu ja sünnikuupäev automaatselt ära.

Ankeedi lõpus peab kandidaat andma oma teadliku nõusoleku, et tulevane tööandja on volitatud tema andmeid edaspidi töötlemata. Ilma nõusolekut andmata ei saa ankeeti ära esitada.

Sulefa

Uue töötaja isikuandmete ankeet

Mari Maasikas

1 Isikuandmed — 2 Kontaktandmed — 3 Pangakonto — 4 Hädaabikontakt — 5 Ülalpeetavad — 6 Kinnitamine

Eesnimi*
Mari

Perekonnanimi*
Maasikas

Isikukood*
48910124473
11 / 11

Sugu*
Naine

Sünnikuupäev*
12.10.1989
pp.kk.aaaa

Edasi

Joonis 11. Uue töötaja isikuandmete ankeet.

9.2 Lahenduse vastavus lähtetingimustele

Üldjoontes võib väita, et loodud lahendus vastab analüüsietapis seatud nõuetele. Funktsionaalsetest nõuetest jäeti välja ainult see, et personalitöötaja saaks personaalsete tööpakkumiste linkide külastamisi kontrollida. Mittefunktsionaalsed nõuded said täidetud.

Valminud lahendus on hõlpsasti liidestatav ka välise personalihaldustarkvaraga Persona V3. Reaalselt toimiv integratsioon Personaga jäeti suure töömahu tõttu siiski lõputöö skoobist välja, kuna käesolev projekt keskendus peamiselt esialgse prototüübi loomisele.

9.3 Kasutajate tagasiside ja võimalused edasiarendusteks

Töö prototüüpi esitleti valitud huvilistele ja viidi läbi piiratud mahus kasutajatestimist, et koguda lõppkasutajate tagasisidet. Kuna loodud veebirakendus on alles prototüübi faasis

ega ole üle avaliku interneti kättesaadav, siis teostati rakenduse esitlemist ja testimist vahetult käesoleva töö autori arvutis.

Tagasisidet koguti vabas vormis, vestluste käigus. Vestlustes osalesid peamiselt Eesti personalivaldkonna spetsialistid. Esmalt tutvustati huvilisele lahendatavat probleemi ja loodud lahendust. Seejärel prooviti juba koos veebirakendust kasutada, olles vaheldumisi erinevates rollides, lahendades mõningaid lihtsaid funktsionaalsetes nõuetes kirjeldatud ülesandeid. Huvilistel paluti hinnata rakenduse arusaadavust, intuiivsust, kasutusmugavust ja kasutajakogemust.

Vestlustelt saadud tagasiside oli üldiselt positiivne ja näidati üles tugevat huvi taolist lahendust enda organisatsioonides proovida. Lahendatav probleem oli kõigile osapooltele arusaadav ja nõustuti lahenduse vajalikkuses. Kõige rohkem meeldis huvilistele see, et kandidaadile tööpakkumise saatmiseks piisab ka ainult tema meiliaadressist. Samuti tõsteti esile lihtsa ja arusaadava ülesehitusega uue töötaja isikuandmete ankeeti.

Lisaks toodi huviliste poolt välja mõningaid puudujääke ning võimalusi edasiarendusteks:

- Personaalse tööpakkumise andmekoosseis võiks olla seadistatav vastavalt organisatsiooni vajadustele. Isegi ühe organisatsiooni sees võivad erinevate ametikohtadega kaasneda erinevad hüved ja vajadus lisada täiendavaid täpsustusi.
- Isikuandmete ankeedi andmekoosseis võiks olla samuti seadistatav vastavalt organisatsiooni vajadustele. Dünaamilise sisuga ankeet võimaldaks tööandja õigustatud huvi korral kandidaadi käest küsida ka muid töötamisega seotud andmeid, mida standardne andmekoosseis hetkel ei toeta. Näiteks auto registreerimisnumber, et uuele töötajale parkimiskoht tellida.
- Isikuandmete ankeet võiks toetada ka kandidaadi foto lisamist, et seda saaks hiljem töötõendil kasutada. Siis ei peaks seda tagantjärele enam töötaja käest eraldi küsima.
- Rakendus võiks olla kasutatav ka vene ja inglise keeles. See suurendaks potentsiaalsete kasutajate ringi märkimisväärselt.

- Isikuandmete ankeet võiks olla liidestatud Rahvastikuregistriga. Kui ankeeti täitev isik ennast mõne riikliku autentimisviisidega autendiks, siis saaks ta ankeedi ette ära täita enda Rahvastikuregistri andmetega. See võimaldaks ankeeti täita kiiremini ja täitmine oleks kandidaadi jaoks mugavam.
- Isikuandmete ankeedil olev elukoha aadress võiks olla liidestatud Eesti keskse aadressiandmete infosüsteemiga (ADS). Siis ei peaks kasutajad enda aadressi vormile käsitsi sisestama, vaid saaksid selle valida eeldefineeritud väärtustega otsingust. See võimaldaks ankeeti täita kiiremini ja vähendaks aadresside valesti kirjutamisel tekkivate vigade hulka.
- Süsteem võiks valideerida isikuandmete ankeedil olevat kontonumbrit (IBAN). See annaks kandidaadile kindlustunde, et ta sisestas enda pangakontonumbri korrektselt.
- Rakendus võiks olla liidestatud Persona personalihaldustarkvaraga, et ei peaks kandidaadi käest kogutud andmeid käsitsi ühest kohast teise tõstma.
- Rakendus võiks olla liidestatud mõne värbamistarkvaraga, et tagada andmete sujuv liikumine üle kogu värbamisprotsessi ja seeläbi vähendada manuaalse töö hulka.

9.4 Majandusliku mõju analüüs

Kui klient võtab kasutusele käesoleva lõputöö raames valminud tarkvara, on saavutatav majanduslik kasu märkimisväärne. Kiireneb värbamisprotsess. Sobilikele kandidaatidele saab tööpakkumisi saata alati ühtses formaadis ja viimistletud kujul. Kandidaatide käest isikuandmete kogumine muutub lihtsamaks, kiiremaks ja töökindlamaks. Andmeid kogutakse süstematiseeritumalt ja väheneb vajadus hiljem käsitsi andmeid ühelt andmekandjalt teisele trükkida.

Kui realiseerida ka eelmises alapeatükis väljatoodud ettepanekud, oleks võimalik lahenduse väärtuspakkumist tõsta veelgi. Huvilistelt saadud tagasiside ja töö autori subjektiivse hinnangu põhjal tundub sedalaadi lahenduse järgi turul olevat praegu võrdlemisi suur nõudlus.

10 Kokkuvõte

Käesoleva lõputöö eesmärgiks oli luua äriklientidele suunatud veebirakendus, mis aitaks personalitöötajatel kandidaatide isikuandmeid paremini hallata, saata kandidaatidele personaalseid tööpakkumisi ning võimaldaks andmete kogumist teostada efektiivsemalt ja süstematiseeritumalt.

Töö tulemusena valmis tervikliku veebirakenduse prototüüp. Lahenduse tööandjapoolne osa võimaldab personalitöötajatel hallata kandidaatide andmeid, saata kandidaatidele personaalseid tööpakkumisi ja kandidaatide käest isikuandmeid koguda. Lahenduse töötajapoolne osa võimaldab kandidaadil veebikeskkonnas tutvuda talle saadetud personaalse tööpakkumisega, tööpakkumisega nõustuda ja täita uue töötaja isikuandmete ankeeti veebivormina.

Huviliste tagasiside loodud lahendusele oli positiivne ja soovitakse seda ka enda organisatsioonides proovida. Antud lahendust edasi arendades ja seda teiste väliste infosüsteemidega liidestades oleks võimalik klientidele pakkuda veelgi rohkem lisandväärtust ja seeläbi kindlustada enda positsioon Eesti personalitarkvarade mitmekesisel maastikul.

Kasutatud kirjandus

- [1] R. Varts ja K. Laurson, „Organisatsiooni strateegia ja personalistrateegia,“ *Personalijuhtimise käsiraamat*, Tallinn, Eesti Personalijuhtimise Ühing PARE, 2012, p. 15.
- [2] R. Varts ja K. Laurson, „Personali värbamine ja valik,“ *Personalijuhtimise käsiraamat*, Tallinn, Eesti Personalijuhtimise Ühing PARE, 2012, p. 97.
- [3] „Fujitsu Eestis,“ Fujitsu Estonia AS, [Võrgumaterjal]. Available: <https://www.fujitsu.com/ee/about/>. [Kasutatud 23 01 2023].
- [4] „Persona avaleht,“ Fujitsu Estonia AS, [Võrgumaterjal]. Available: <https://persona.fujitsu.ee/Persona/Avaleht/>. [Kasutatud 16 02 2023].
- [5] V. Jelov ja K. Laas, „Eesti personalitarkvara turuuuring 2017. a.,“ *Personalivaldkonna infosüsteemid, sissejuhatus: olevik, tulevik, rakendamine ja valik*, Tallinn, HRConf, 2017, p. 280.
- [6] V. Jelov ja K. Laas, „Värbamine ja kandidaatide seire,“ *Personalivaldkonna infosüsteemid, sissejuhatus: olevik, tulevik, rakendamine ja valik*, Tallinn, HRConf, 2017, p. 110.
- [7] Council of the EU and the European Council, „The general data protection regulation,“ Council of the EU and the European Council, [Võrgumaterjal]. Available: <https://www.consilium.europa.eu/en/policies/data-protection/data-protection-regulation/>. [Kasutatud 11 04 2023].
- [8] Tööinspektsioon, „Töötaja privaatsus,“ Tööinspektsioon, 08 05 2021. [Võrgumaterjal]. Available: <https://www.tooelu.ee/et/69/tootaja-privatsus>. [Kasutatud 11 04 2023].
- [9] Isikuandmete kaitse üldmäärus, „Mõisted,“ Isikuandmete kaitse üldmäärus, [Võrgumaterjal]. Available: <https://gdprinfo.eu/et/et-article-4>. [Kasutatud 11 04 2023].
- [10] Riigi Teataja, „Töölepingu seadus,“ Riigi Teataja, 01 97 2009. [Võrgumaterjal]. Available: <https://www.riigiteataja.ee/akt/107032023049>. [Kasutatud 11 04 2023].
- [11] General Data Protection Regulation, „Nõusoleku andmise tingimused,“ Isikuandmete kaitse üldmäärus, [Võrgumaterjal]. Available: <https://gdprinfo.eu/et/et-article-7>. [Kasutatud 11 04 2023].
- [12] Isikuandmete kaitse üldmäärus, „Töötlemise turvalisus,“ Isikuandmete kaitse üldmäärus, [Võrgumaterjal]. Available: <https://gdprinfo.eu/et/et-article-32>. [Kasutatud 11 04 2023].
- [13] Riigi Tugiteenuste Keskus, „Riigitöötaja iseteenindusportaal,“ Riigi Tugiteenuste Keskus, [Võrgumaterjal]. Available: <https://www.rtk.ee/riigitootaja-iseteenindusportaal>. [Kasutatud 10 04 2023].
- [14] Riigi Tugiteenuste Keskus, „Riigitöötaja iseteenindusportali kasutusjuhendid,“ Riigi Tugiteenuste Keskus, [Võrgumaterjal]. Available:

- <https://www.rtk.ee/riigitootaja-iseteenindusportaal/rtip-juhendid>. [Kasutatud 10 04 2023].
- [15] „RecruitLab,“ Crunchbase, [Võrgumaterjal]. Available: <https://www.crunchbase.com/organization/recruitlab>. [Kasutatud 10 04 2023].
- [16] Recruitment Software OÜ, „Meist,“ Recruitment Software OÜ, [Võrgumaterjal]. Available: <https://recruitlab.co.uk/et/recruitlabist/>. [Kasutatud 10 04 2023].
- [17] Recruitment Software OÜ, „Funktsioonid,“ Recruitment Software OÜ, [Võrgumaterjal]. Available: <https://recruitlab.co.uk/et/funktsioonid/>. [Kasutatud 10 04 2023].
- [18] Google, „Ülevaade,“ Google, [Võrgumaterjal]. Available: <https://www.google.com/forms/about/>. [Kasutatud 10 04 2023].
- [19] A. Wasserman, „Software Entropy Explained: Causes, Effects, and Remedies,“ Toptal, [Võrgumaterjal]. Available: <https://www.toptal.com/software/software-entropy-explained>. [Kasutatud 14 04 2023].
- [20] P. Walpita, „Software Architecture Patterns – Which one to choose?,“ Medium, 09 07 2019. [Võrgumaterjal]. Available: <https://priyalwalpita.medium.com/software-architecture-patterns-which-one-to-choose-1e368b43fe70>. [Kasutatud 14 04 2013].
- [21] Z. Gollwitzer, „Introduction to Software Architecture (Monolithic vs. Layered vs. Microservices),“ DEV Community, 18 10 2020. [Võrgumaterjal]. Available: <https://dev.to/zachgoll/introduction-to-software-architecture-monolithic-vs-layered-vs-microservices-452>. [Kasutatud 14 04 2024].
- [22] Talend, „Monolithic vs. Microservices: a guide to application architecture,“ Talend, [Võrgumaterjal]. Available: <https://www.talend.com/resources/monolithic-architecture/>. [Kasutatud 14 04 2023].
- [23] E. G. Gallardo, „What Is a Monolithic Application?,“ Built In, 28 12 2022. [Võrgumaterjal]. Available: <https://builtin.com/software-engineering-perspectives/monolithic-application>. [Kasutatud 14 04 2023].
- [24] „Multitenancy,“ Gartner, [Võrgumaterjal]. Available: <https://www.gartner.com/en/information-technology/glossary/multitenancy>. [Kasutatud 16 02 2023].
- [25] L. Gupta, „REST Resource Naming Guide,“ 30 12 2022. [Võrgumaterjal]. Available: <https://restfulapi.net/resource-naming/>. [Kasutatud 16 02 2023].
- [26] C. Dulanga, „REST vs GraphQL vs gRPC,“ Bits and Pieces, 10 07 2022. [Võrgumaterjal]. Available: <https://blog.bitsrc.io/rest-vs-graphql-vs-grpc-684edfacf810>. [Kasutatud 14 04 2023].
- [27] L. Gupta, „What is JSON,“ HowToDoInJava, 27 09 2023. [Võrgumaterjal]. Available: <https://restfulapi.net/introduction-to-json/>. [Kasutatud 14 04 2023].
- [28] „What is Entity Framework?,“ Entity Framework Tutorial, [Võrgumaterjal]. Available: <https://www.entityframeworktutorial.net/what-is-entityframework.aspx>. [Kasutatud 16 02 2023].
- [29] „What is LINQ?,“ TutorialTeacher, [Võrgumaterjal]. Available: <https://www.tutorialsteacher.com/linq/what-is-linq>. [Kasutatud 16 02 2023].
- [30] „Projection operations (C#),“ Microsoft, 10 06 2022. [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/projection-operations>. [Kasutatud 16 02 2023].

- [31] K. Abuhakmeh, „Entity Framework Core 5 – Pitfalls To Avoid and Ideas to Try,“ JetBrains, 24 02 2021. [Võrgumaterjal]. Available: <https://blog.jetbrains.com/dotnet/2021/02/24/entity-framework-core-5-pitfalls-to-avoid-and-ideas-to-try/>. [Kasutatud 16 02 2023].
- [32] „What is Code-First?,“ Entity Framework Tutorial, [Võrgumaterjal]. Available: <https://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx>. [Kasutatud 16 02 2023].
- [33] F. Lasserre, „The table naming dilemma: singular vs. plural,“ Medium, 25 01 2027. [Võrgumaterjal]. Available: <https://medium.com/@fbnlr/the-table-naming-dilemma-singular-vs-plural-dc260d90aaff>. [Kasutatud 16 02 2023].
- [34] P. Rospel, „Andmemudelite normaliseerimine,“ IT Kolledž, 20 09 2021. [Võrgumaterjal]. Available: <https://enos.itcollege.ee/~priit/1.%20Andmebaasid/1.%20Loengumaterjalid/07/7.htm>. [Kasutatud 16 02 2023].
- [35] M. Metshein, „Andmete modelleerimine: normaliseerimine,“ Haapsalu Kutsehariduskeskus, [Võrgumaterjal]. Available: <https://www.metshein.com/unit/andmete-modelleerimine-normaliseerimine/>. [Kasutatud 16 02 2023].
- [36] „Don't Repeat Yourself (DRY),“ Principles Wiki, 18 10 2021. [Võrgumaterjal]. Available: http://principles-wiki.net/principles:don_t_repeat_yourself. [Kasutatud 16 02 2023].
- [37] „Modeling Bridge Tables,“ Oracle, [Võrgumaterjal]. Available: https://docs.oracle.com/middleware/12212/biee/BIEMG/GUID-3F5B2C9F-9FE3-48F7-A081-07241F6FBDC1.htm#modeling_bridge. [Kasutatud 16 02 2023].
- [38] „Frontend vs Backend,“ 06 01 2023. [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/frontend-vs-backend/>. [Kasutatud 16 02 2023].
- [39] „CLI Overview and Command Reference,“ [Võrgumaterjal]. Available: <https://angular.io/cli>. [Kasutatud 16 02 2023].
- [40] G. Singhal, „Why Do We Need Single-page Applications?,“ Pluralsight, 09 04 2020. [Võrgumaterjal]. Available: <https://www.pluralsight.com/guides/why-do-we-need-a-single-page-application>. [Kasutatud 12 04 2023].
- [41] Google, „Introduction to components and templates,“ Google, [Võrgumaterjal]. Available: <https://angular.io/guide/architecture-components>. [Kasutatud 13 04 2023].
- [42] Google, „Introduction to Angular concepts,“ Google, [Võrgumaterjal]. Available: <https://angular.io/guide/architecture>. [Kasutatud 16 02 2023].
- [43] D. Saha, „Feature Modules in Angular – Why Required and When to Use?,“ DZone, 01 12 2020. [Võrgumaterjal]. Available: <https://dzone.com/articles/feature-modules-in-angular-why-required-and-when-t>. [Kasutatud 13 04 2023].
- [44] Google, „Introduction to services and dependency injection,“ Google, [Võrgumaterjal]. Available: <https://angular.io/guide/architecture-services>. [Kasutatud 13 04 2023].
- [45] Google, „Angular Routing,“ Google, [Võrgumaterjal]. Available: <https://angular.io/guide/routing-overview>. [Kasutatud 08 04 2023].

- [46] Google, „HTTP - Intercept requests and responses,“ Google, [Võrgumaterjal]. Available: <https://angular.io/guide/http-intercept-requests-and-responses>. [Kasutatud 13 04 2023].
- [47] „Angular Material Tutorial,“ [Võrgumaterjal]. Available: https://www.tutorialspoint.com/angular_material/index.htm. [Kasutatud 16 02 2023].
- [48] „Material Design,“ Google, [Võrgumaterjal]. Available: <https://m2.material.io/design>. [Kasutatud 16 02 2023].
- [49] „CSS Preprocessor SASS,“ GeeksforGeeks, [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/css-preprocessor-sass/>. [Kasutatud 16 02 2023].
- [50] „Advantages and Disadvantages of Sass,“ Javatpoint, [Võrgumaterjal]. Available: <https://www.javatpoint.com/sass-advantages-and-disadvantages>. [Kasutatud 16 02 2023].
- [51] J. Watmore, „.NET 6.0 - JWT Authentication with Refresh Tokens Tutorial with Example API,“ Jason Watmore's Blog, 24 01 2022. [Võrgumaterjal]. Available: <https://jasonwatmore.com/post/2022/01/24/net-6-jwt-authentication-with-refresh-tokens-tutorial-with-example-api>. [Kasutatud 17 02 2023].
- [52] G. Peipman, „Sissejuhatus testimisse,“ [Võrgumaterjal]. Available: <https://kool.gunnarpeipman.com/programmeerimine-2/testimine/>. [Kasutatud 10 04 2023].
- [53] S. Saurav, „The exponential cost of fixing bugs,“ DZone, 07 07 2021. [Võrgumaterjal]. Available: The Exponential Cost of Fixing Bugs. [Kasutatud 11 04 2023].
- [54] IBM, „What is software testing?,“ IBM, [Võrgumaterjal]. Available: <https://www.ibm.com/topics/software-testing>. [Kasutatud 10 04 2023].
- [55] JavaTpoint, „Advantages and Disadvantages of Manual Testing,“ JavaTpoint, [Võrgumaterjal]. Available: <https://www.javatpoint.com/advantages-and-disadvantages-of-manual-testing>. [Kasutatud 11 04 2023].
- [56] SmartBear Software, „What Is Swagger?,“ SmartBear Software, [Võrgumaterjal]. Available: <https://swagger.io/docs/specification/about/>. [Kasutatud 10 04 2023].
- [57] Postman, „What is Postman?,“ Postman, [Võrgumaterjal]. Available: <https://www.postman.com/product/what-is-postman/>. [Kasutatud 10 04 2023].
- [58] .NET Foundation, „About xUnit.net,“ .NET Foundation, [Võrgumaterjal]. Available: <https://xunit.net/>. [Kasutatud 10 04 2023].
- [59] Ethereal Email, „Ethereal,“ Ethereal Email, [Võrgumaterjal]. Available: <https://ethereal.email/>. [Kasutatud 11 04 2023].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Kristo Palo

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Veebirakendus uute töötajate isikuandmete haldamiseks“, mille juhendaja on Meelis Antoi
 - 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.

15.05.2023

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 – Olemi-suhte diagramm

