

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Priit Post

**Veebipõhise tagarakendi arendus pääsla
läbipääsu infosüsteemi näitel**

Bakalaureusetöö

Juhendaja: Toomas Lepikult
PhD

Tallinn 2021

Autorideklaratsioon

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

Autor: Priit Post

17.05.2021

Annotatsioon

Käesolev lõputöö käsitleb vangla turvalisuse tõstmisega seonduvat. Täpsemalt puudutab lõputöö vanglate pääslate tegevust nii inimeste, sõidukite kui ka inimestega kaasas olevate tehnikavahendite registreerimisel vanglasse sisenemisel ja vanglast väljumisel. Täna toimub inimeste, sõidukite ja tehnikavahendite lubade menetlus enamasti e-kirja teel ning load prinditakse välja või arhiveeritakse võrgukettal ja nende pääsla läbimine registreeritakse MS Exceli tabelis. Nimetatud tööriistade kasutamine on ajamahukas ja selle veamäär on võrdlemisi kõrge. Probleemi lahendamiseks näeb autor vajadust ühe keskse infosüsteemi järele, mis oleks kõigi kolme vangla peale ühine. Töö mahtu kuulub sellest vaid tagarakenduse arendus. Ülejäänud osa jääb arendamiseks väljapoole käesoleva töö mahtu.

Lähtuvalt eelmises lõigus toodud ülesandepüstitusest valis autor välja kasutajate jaoks sobivaima platvormina veebirakenduse. Toimiva infosüsteemi abil on võimalik inimesi, sõidukeid ja inimesega kaasas olevaid tehnikavahendeid nii sisenemisel kui ka väljumisel registreerida, hallata vanglasse sisenemise lubasid ning pärida rakenduse vahendusel nii hetkeseisu kui ka soovitud ajavahemiku kohta andmebaasist andmeid.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 30 leheküljel, kuut peatükki, viit joonist.

Abstract

Development of the Web-based Backend Application for Entrance Information System

The present thesis deals with the improvement of prison security. More specifically, the thesis concerns the activities of prison entrances for the registration of people, vehicles and technical equipment accompanying people when entering and leaving prison. Today, the procedure for people, vehicles and technical equipment is usually carried out by e-mail, and permits are printed or archived on a network drive and their passage is recorded in an MS Excel spreadsheet. The use of these tools is time consuming and has a relatively high error rate. To solve the problem, the author sees the need for one central information system, which would be common to all three prisons. The scope of the work includes only the development of the back application. The rest is outside the scope of this work.

With the help of a functioning information system, it is possible to register people, vehicles and accompanying technical equipment both on entry and exit, manage prison entry permits and query the database for data on the current moment and the desired period via the application.

Based on the task set out in the first paragraph, the author selected a web application as the most suitable platform for users. The server side or backend application is developed in the C # programming language and uses its .NET framework and ASP.NET for the web application development. The code structure of the backend is layered. The Entity Framework Core was used as the ORM model. REST was preferred for API design.

The pre-application is planned to be developed using the Vue.js framework and the TypeScript programming language. In addition to this front-end application, the system under development must be integrated with hardware components, such as an identity metadata reader, a camera for face recognition and document photo comparison, and, if necessary, an access control system.

The thesis is in Estonian and contains 30 pages of text, six chapters, five figures.

Lühendite ja mõistete sõnastik

3D	Kolme-dimensiooniline
AD	Identiteediteenus <i>Windows</i> platvormile (<i>Active Directory</i>)
API	Rakenduse programmeerimise liides (<i>Application Programming Interface</i>)
C#	Programmeerimiskeel (<i>C Sharp</i>)
DBMS	Andmebaasi juhtsüsteem (<i>Database Management System</i>)
DTO	Klass erinevate rakenduse kihtide vahel andmete teisendamiseks (<i>Data Transfer Object</i>)
EF Core	Raamistik (<i>Entity Framework Core</i>)
IT	Infotehnoloogia
JVK	Justiitsministri 05.09.2011 määrus nr 44 „Järelevalve korraldus vanglas“
KIR	Kinnipeetavate isikute register
LINQ	Programmeerimiskeelde integreeritud andmebaasipäringu raamistik (<i>Language Integrated Query</i>)
MacOS	Ettevõtte <i>Apple</i> seadmetele paigaldatav operatsioonisüsteem
RIK	Registrite ja Infosüsteemide Keskus
MS	Ettevõtte <i>Microsoft</i>
OAuth	Kolmandate osapoolte pakutav autentimise teenus
ORM	Tööriist, mida kasutatakse rakenduse andmetele juurdepääsu loogika abstraheerimiseks (<i>Object-Relational Mapping</i>)
REST	Tehnoloogia nimi veebirakenduse loomiseks (<i>Representational State Transfer</i>)
SAP	Personali ja raamatupidamise haldamise tarkvara
SOAP	Lihtne objektile juurdepääsu protokoll (<i>Simple Object Access Protocol</i>)
SQL	Andmebaasi keel (<i>Structured Query Language</i>)
URL	Ühtne ressursi lokaator ehk Interneti aadress (<i>Uniform Resource Locator</i>)
VangS	Vangistusseadus
VPN	Privaatne virtuaalne arvutivõrk (<i>Virtual Private Network</i>)
VSKE	Justiitsministri 30.11.2000 määrus nr 72 „Vangla sisekorraeskiri“
XML	Laiendatav märgistuskeel (<i>Extensible Markup Language</i>)

Sisukord

Jooniste loetelu	7
1 Sissejuhatus	8
2 Probleemi kirjeldus.....	10
3 Võimalikud lahendused.....	12
3.1 Turul saadaolevad lahendused.....	12
3.2 Probleemi lahendus.....	12
3.3 Õiguslikud nõuded.....	13
3.4 Äriliste nõuete üldine kirjeldus.....	14
3.5 Äriliste nõuete kirjeldus MoSCoW mudeli järgi.....	16
3.6 Äriprotsessi lihtsustatud skeem	20
3.7 Olemi-suhte diagramm	22
4 Arendus.....	24
4.1 RIK-iga seotud tehnilised nõuded	24
4.2 Arenduskeele valik	25
4.3 Rakenduse platvormi valik	26
4.4 Raamistike valik	29
4.4.1 Üldine raamistik	29
4.4.2 Veebiraamistik.....	29
4.4.3 ORM raamistik	30
4.5 API disain	30
4.6 Autentimine	32
4.7 Rakenduse arhitektuur	33
4.8 Dokumentatsioon.....	34
5 Töö tulem.....	35
6 Kokkuvõte	36
Kasutatud kirjandus	37
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	38

Jooniste loetelu

Joonis 1. Äriprotsessi lihtsustatud skeem.	21
Joonis 2. Pääsla infosüsteemi olemi-suhte diagramm.	23
Joonis 3. Kahe andmeväljaga SOAP päringu näidis	31
Joonis 4. Kahe andmeväljaga REST päringu näidis.....	31
Joonis 5. Rakenduse arhitektuuriline skeem.	34

1 Sissejuhatus

Eestis on kolm vanglat, mis asuvad Tallinnas, Tartus ja Jõhvis. Vangla territooriumil asuvad hooned ja rajatised, mis paiknevad nii vangla piirde sees kui ka sellest väljaspool. Väljaspool vangla territooriumit asuvad reeglina: avavanglaosakonna hoone, vangla tehnohooned ning koerte majutamiseks ettenähtud hoone ja selle juures paiknev rajatis koerte treenimiseks. Vangla piirdest väljapoole jäävatele hoonetele on füüsiline ligipääs kõigil sõltumata sellest, et nendesse sisenemiseks on tarvis omada vastavat õigust ja füüsilist võtit või kaarti. Vangla piirdes sees paiknevad hooned, millel on järgmine otstarve:

- Vangide majutamine
- Toidu valmistamine
- Sportimine
- Vangide tööhõive, õppimine ja taasühiskonnastavate tegevuste läbiviimine
- Meditsiiniteenuse osutamine
- Vangide kokkusaamised
- Muud tööruumid

Loetelus toodud hoonetele ligipääsemiseks on tarvis siseneda vangla territooriumile läbi vangla piirdel oleva pääslahoone. Pääsla on ruum või selle mõtteline osa, mille kaudu isikud ja transpordivahendid vanglasse sisenevad ja vanglast väljuvad. Pääsla koosneb sõidukite liikumiseks ette nähtud osast, isikute liikumiseks ette nähtud osast ja pääslas töötavate ametnike tööruumidest. Vanglasse sisenetakse ja sealt väljutakse ainult läbi pääsla [1]. Selleks, et vanglasse pääseda, peab olema inimesel ja temaga kaasas olevatele asjadele vastav luba. Kui siseneval inimesel on vastav luba olemas, siis vangla territooriumile sisenemine registreeritakse pääslas vanglaametniku poolt. Nimetatud registreerimistoimingu ning lubade menetlemise ja haldamiseks puudub sobiv tehniline

lahendus. See kätkeb endas mitmeid vangla turvalisusega seotud riske, mis teeb käesoleva bakalaureuse töö teema aktuaalseks.

Töö koosneb kuuest peatükist, millest esimene on sissejuhatus ja kuues kokkuvõte. Töö täpsem probleematika on avatud teises peatükis pealkirjaga „Probleemi kirjeldus“, kus tuuakse välja, milline on tänane vangla pääsla töökorraldus ning millised on selle puudused.

Töö kolmandas peatükis pakutakse välja võimalik lahendus koos selle funktsionaalsete nõuetega.

Töö neljandas peatükis analüüsitakse infosüsteemi arendusega seotud valikuid.

Töö viiendas peatükis võetakse kokku lõputöö tulemused.

Käesoleva bakalaureuse töö eesmärgiks on luua vanglatele ja justiitsministeeriumi vanglate osakonnale ühtne infosüsteem. Käesolevasse bakalaureuse töösse mahub üksnes vanglate pääsla infosüsteemi tagarakenduse osas, milles see katab ära tänase tühimiku minimaalses vajalikus osas. Rakenduse edasised arendusvajadused on kirjeldatud nõuetena punktis 3.3 toodud MoSCoW mudelis.

2 Probleemi kirjeldus

Eesti vanglate turvalisus on üheks kogu Eesti ühiskonna turvalisuse komponendiks. Kui kurjategija saadetakse vanglasse karistust kandma, siis eeldatakse, et ta seal ka püsib kuni vabanemiseni. See on Eesti karistuspoliitika toimimise üks olulisemaid komponente.

Üheks vangla turvalisuse alustalaks on toimiv pääsla. Tähtis roll on kindlasti ametnikel, kes peavad veenduma, et vanglasse sisenevad ja sealt väljuvad vaid need inimesed, kellel on selleks vastav luba. Kui sõnastada see teisiti, siis ametnikud peavad veenduma, et vanglast ei üritaks väljuda mõni vang, kellel selleks õigust ei ole või vanglasse ei üritaks siseneda inimene, kellel on plaanis seada ohtu vangla julgeolekut. Pääslas töötavad ametnikud peavad lisaks läbi otsima nii vanglat külastavad inimesed kui ka sisenevad ja väljuvad sõidukid veendumaks, et vangla territooriumile ei jõuaks inimesi ega esemeid, mis on vanglas keelatud või mille kohta ei ole vanglateenistus väljastanud luba, ja territooriumilt ei väljuks inimesi, kes selleks luba ei oma.

Vanglad peavad täna sisenevate ja väljuvate inimeste ning sõidukite ja nendega kaasas olevate tehnikavahendite üle arvestust MS Exceli töövihikutes. See tähendab, et puudub ühtne vanglateülene infosüsteem. Töövihikute pidamise suurteks miinusteks on:

- Pole võimalik saada kiiret ülevaadet vanglas viibivatest inimestest ja sõidukitest. See võib kriisiolukorras seada ohtu kellegi elu või tervise.
- Vanglal puudub teave teise vangla tehtud otsuse kohta kellegi vanglasse lubamise või mittelubamise kohta, mis võib põhjustada vanglatevahelist ebaühtlast halduspraktikat. Sellist praktikat on halduskohtus keeruline kaitsta, kuna isiku vanglasse mitte lubamise põhjused peavad erinevates vanglates olema samasugused.
- MS Exceli töövihikutega on keeruline pidada arvestust isiku- või esemepõhiselt. See tähendab, et töövihikud ei võimalda automaatselt tekitada isikupõhist ülevaadet kõigist vanglakülastustest. Esemetega seotud problemaatika on sarnane – täna puudub vanglateülene esemete- või sõidukitepõhine ülevaade nende

erinevates vanglates sisenemiste kohta. Seetõttu võivad jääda võimalikud probleemkohad või riskid vanglaametnikele märkamata.

- Tänaused töövihikud ei võimalda saada mõistliku kiirusega statistilisi andmeid. Näiteks, kui palju on mingil perioodil külastanud üht vanglat inimesi või sõidukeid; kui palju on külastanud erinevaid vanglaid mingid kindlad inimesed – nt kaitsjad või vangide lähedased seoses kokkusaamisega. See raskendab erinevate juhtimisotsuste tegemist.

Puuduv korralik infosüsteem võib põhjustada vanglate pääslates inimlikke vigu nt töövihikusse sissekannete tegemisel. Lisaks ei võimalda kasutusel olevad lahendused anda kriisiolukorras sündmust juhtivale ametnikule piisava kiirusega kriitilist informatsiooni, mida pääslasse on kogutud. Loetletud probleemid võivad tõsiselt mõjutada nii vanglat külastavate isikute kui ka kogu ühiskonna turvalisust.

Arendatav infosüsteem tuleb nõ tühjale kohale – see asendab üksnes MS Exceli töövihikuid. Vanglates on infosüsteemi kasutusele võtmiseks olemas vajalik infrastruktuur – lõppkasutaja seadmed (PC) ja kesksed serverid, mis teenindavad vanglaid. Kogu arendusvajadust silmas pidades tuleb soetada juurde isikut tõendavate dokumentide metaandmeid lugevaid seadmeid ja kaameraid, millega on võimalik teha fotosid inimestest, kes vanglasse sisenevad.

3 Võimalikud lahendused

Käesolevas peatükis antakse ülevaade kehtivatest õiguslikest nõuetest ning äriolistest nõuetest, mille piires tuleb loodav infosüsteem arendada. Punktis 3.1 tuuakse välja õiguslikud nõuded, punktis 3.2 ärioliste nõuete üldine kirjeldus ning punktis 3.3 tuuakse välja MoSCoW mudeli järgi määratletud nõuded. Punktis 3.3 toodud nõuetest on osa, mis mahuvad käesoleva töö mahtu kui ka need, mis sinna ei mahu ja kuuluvad arendamisele selle töö jätkuna.

3.1 Turul saadaolevad lahendused

Täna ei ole turul sellist toodet, mis õnnestuks kasutusele võtta ilma täiendavate suurte arendusteta. Turg on selles valdkonnas olematu ja pigem on sarnaseid süsteeme üritatud ise arendada. Sellist teed on läinud ka Eesti Kaitsevägi. Kahjuks on nende soovid vanglateenistuse omadest erinevad, mistõttu ei ole nendega ühise infosüsteemi loomine otstarbekas. Nimelt soovivad nemad arendada välja tarkvara, mille soovivad kasutusele võtta igas väeosas eraldi ning ühtset loa menetlemise ja haldamise süsteemi ei plaani nad sellisel kujul välja arendada. Lisaks on vanglateenistusel vajadus kõiksugu tehnikavahendite jms üle eraldi arvestust pidada ning plaan välja arendada võimekus ka kaamerate abil inimeste ja sõidukite tuvastamise kohta.

3.2 Probleemi lahendus

Esimene samm andmebaasi kujundamisel on nõuete kogumine ja analüüs. Selle etapi käigus küsitlevad andmebaasi disainerid potentsiaalseid andmebaasi kasutajaid, et neist aru saada ja dokumenteerida andmenõuded. Selle sammu tulemuseks on lühidalt koostatud kasutajate nõuded [2]. Kuivõrd autor on ise käesoleva probleemi lahendamiseks loodud andmebaasi tellijaks, siis töö käigus küsitlust läbi ei viidud, vaid lähtuti autori hinnangutest.

3.3 Õiguslikud nõuded

Rakenduses kasutatakse üksnes eesti keelt, kuna vastavalt keeleseadusele on Eesti riigikeeleks eesti keel ja kõik vanglaametnikud on kohustatud valdama eesti keelt tasemel C1, mis on kõrgtase.

Vanglate järelevalve täpsema korra kehtestab valdkonna eest vastutav minister [3]. Vangistuse valdkonda reguleerivad küsimused on justiitsministri pädevuses [4]. Justiitsminister reguleerib vangla peaaeslaga seonduvaid nõudeid JVK kolmandas peatükis. Osa peaaesla tegevust reguleerivaid nõudeid asuvad ka peatükkides 4 ja 6¹.

Vanglasse siseneva ja vanglast väljuva isiku kohta registreeritakse temaga kaasas oleva tehnilise vahendi andmed. Kui vanglasse sisenev või vanglast väljuv isik ei tööta selles vanglas, registreeritakse tema: ees- ja perekonnanimi, ametiasutuse nimetus tööalase külastuse korral, saabumise põhjus, vanglasse sisenemise aluseks oleva dokumendi number, vanglasse sisenemise ja vanglast väljumise kuupäev ning kellaaeg ning kaasas oleva tehnikavahendi andmed vanglasse sisenemisel ja vanglast väljumisel. Andmeid vanglasse sisenevate isikute kohta säilitatakse kolm aastat [5]. Sarnaselt sisenemisele tuleb registreerida ka vanglast väljumine. Väljumisel tuleb kontrollida nii isikut kui ka temaga kaasas olnud vahendeid.

Isiku vanglasse sisenemine registreeritakse ja sisenemise aluseks olev dokument võetakse vanglas viibimise ajaks hoiule [5]. Sisenemise aluseks olevate dokumentide loetelu on samuti reguleeritud eelpool nimetatud määrusega. Nimelt, isik siseneb vanglasse, kas töötõendi või liikmetunnistuse alusel, alalise sissepääsu loa alusel, lühiajalise sissepääsuloa alusel koos isikut tõendava dokumendiga või kokkusaamise graafiku ja isikut tõendava dokumendi alusel [5].

Töötõendi alusel on vanglasse lubatud siseneda järgmistel isikul: vanglateenistuja, justiitsministeeriumi ametnik ja töötaja, kohtunik, prokurör, Eesti Advokatuuri liige, politseiametnik, päästeametnik, Maksu- ja Tolliameti ametnik, notar, Õiguskantsleri Kantselei teenistuja, Registrite ja Infosüsteemide Keskuse töötaja [5]. Alalise sissepääsu loa üle otsustab kas justiitsministeerium või vangla direktor [5]. Lühiajalise sissepääsu loa annab vangla direktor või tema määratud ametnik [5]. Lühiajalise kokkusaamise üle otsustab vangi inspektor-kontaktisik [6].

Üldjuhul vangile keelatud esemeid vanglasse kaasa võtta ei saa. Siiski esinevad erandid, kuivõrd vahel on tarvis vanglasse sisenevatel isikutel tööülesannete täitmise eesmärgil võtta kaasa ka vangile keelatud esemeid. JVK §-s 21 on välja toodud regulatsioon selle kohta, kes ja milliste lubade olemasolul või tingimuste esinemisel on õigustatud vanglasse kaasa võtma tehnilisi vahendeid, mis on vangidele keelatud.

Sõidukite vanglasse sisenemisel kohaldatakse sarnast regulatsiooni isikutele. See tähendab, et sõiduki sisenemiseks vangla territooriumile peab olema eelnev luba.

Sisenemisel erakorralise sündmuse korral vanglasse kehtivad samuti erisused. Vanglas erakorralise sündmuse lahendamiseks sisenevad isikud ja sõidukid vanglasse erakorralise sündmuse lahendamise juhi korralduse alusel ja viisil. Sisenemise ja väljumise kohta registreeritakse: isikute arv ja nimed, sõiduki liik, mark ja registreerimisnumber, kuupäev ja kellaeg ja vanglasse sisenemise põhjus. Andmed registreeritakse vanglas viibimise jooksul [5]. See tähendab, et loetletud andmeid ei tule registreerida nende isikute vanglasse viibimise hetkel, vaid tuleb registreerida sisuliselt nende väljumise hetkel.

3.4 Äriliste nõuete üldine kirjeldus

Iga vanglasse siseneva isiku, sõiduki või tehnikavahendi jaoks peab olema sisenemiseks vajalik luba või õigusaktist tulenev alus. Teatud isikute puhul määratleb JVK nende vanglasse sissepääsemise tingimused – nt on teatud isikutel lubatud automaatselt vanglasse siseneda (selle vangla töötajad, justiitsministeeriumi ametnikud jt) või on teatud isikutel luba võtta vanglasse kaasa vabalt valitud tehniline vahend. Teiste isikute puhul, keda õigusaktid ei nimeta, peab olema vanglasse sisenemiseks vangla direktori või tema määratud ametniku luba. Pääsلاس töötavale ametnikule peab info nii isikute lubamise kui ka nendega kaasas olevate esemete kohta olema läbi loodava infosüsteemi teada.

Kui isik siseneb vanglasse, siis kontrollib pääsلاس ametnik, kas tal on õigus vanglasse siseneda (see tähendab, et isikul kas õigusaktist tulenev või vangla direktori või tema määratud ametniku luba) ning kas talle on lubatud kaasa võtta ka tehnikavahend või sõiduk. Olulised on ka loa kehtivuse kuupäevad, kuna peaaegu ükski vanglasse sisenemise või esemete vanglasse lubamisega seotud luba ei ole tähtjatu.

Juhul kui isikul on vanglasse sisenemise luba ja vajadusel ka tehnikavahendi luba olemas, registreerib pääsla ametnik isiku vanglasse sisse. Vastupidisel juhul saadab pääsla ametnik isiku vanglast välja. Vanglasse registreeritakse ka isikuga kaasas olev tehnikavahend ja sõiduk. Isiku vanglast väljumisel veendub pääsla ametnik, et isik on seesama, kes vanglasse sisenes, ja et tal on kaasas need samad tehnikavahendid, millega ta vanglasse sisenes. Juhul kui kõik on korrektne ja vastab lubade tingimustele, registreeritakse isik vanglast välja.

Sama kehtib ka tehnikavahendile ja sõidukile, kuid erisusega, et tehnikavahendit ega sõidukit ei registreerita kunagi vanglasse sisse ega sealt välja ilma juhita või valdajata. Sõiduk siseneb vanglasse ainult koos juhiga. See tähendab, et kaasreisijad peavad sisenema vanglasse läbi isikute pääsla. Mõistagi on siin erisusi, näiteks:

- Vangide transportimiseks kasutatav konvoibuss võib siseneda vanglasse koos ametnike ja vangidega.
- Vanglasse erakorralist sündmust lahendama asuvad vanglate relvastatud üksuse liikmed võivad siseneda vanglasse nii jalgsi kui ka sõidukis, nii üksi kui ka grupis.
- Muude eritalituse sõidukite (nt kiirabi ning päästeameti auto) reageerimisel vanglas toimuvale erakorralisele sündmusele ei lasta sõidukis olevatel inimestel siseneda vanglasse isikute pääsla kaudu, vaid nad sisenevad vangla territooriumile sõidukis.

Vanglasse sisenemise luba antakse, kas ühele inimesele või korraga mitmele. Sõidukile antud loaga on alati seotud inimene, välja arvatud vangla enda sõidukid, mille puhul võivad vanglasse nendega siseneda kõik vangla töötajad. Luba antakse isikule sisenemiseks kas mitte ühegi, ühe või mitme tehnikavahendiga, kuid isik võib realselt vanglasse siseneda kas loal toodud maksimaalse tehnikavahendite hulgaga või vähemaga – see tähendab, et kui isik vanglasse siseneb, ei pea tal olema kaasas ühtegi vahendit, milleks talle on luba antud.

Vanglasse sisenemise aluseks olevad load on erinevad. Nagu mainitud, võivad isikud siseneda justiitsministri määruse alusel. Lisaks võivad nad siseneda vangla direktori või tema määratud ametniku ühekordse otsuse alusel. Otsusega antav luba on kas lühiajaline või alaline. Alaline luba tähendab siiski sissepääsuõigust maksimaalselt kuni üheks

aastaks. Ühekordse loa kehtivus saab olla maksimaalselt üks kuu. Lisaks võib vanglasse sisenemise aluseks olla ka kokkusaamiste graafik, mis koosneb erinevate ametnike otsustest enda vangide osas, vangide saatmisega seotud dokumendid (väljasõidutunnistus, saatmise korraldus või saatenimekiri) või kohtuotsus või -määrus, millega pöörati vangi karistus täitmisele.

3.5 Äriliste nõuete kirjeldus MoSCoW mudeli järgi

MoSCoW mudel on üks võimalik viis, kuidas infosüsteemi nõudeid prioritseerida. MoSCoW on poolenisti akronüüm, mis tuleneb inglisekeelsetest sõnadest „*must*“, „*should*“, „*could*“ ja „*would*“. Ehk eesti keelses vasted oleksid neile „kohustuslikud nõuded“, „nõuded, mida peaks täitma“, „nõuded, mida võiks täita“ ja „nõuded, mida ei peaks täitma“.

MoSCoW meetod kinnitab, et kõik nõuded on hädavajalikud, kuid need tuleks tellida nii, et need annaksid varakult kõige olulisemaid ja otsemaid äritulusid. Nõuded on jaotatud ühte neljast kategooriast: peab olema, soovituslikud, nõuded, mis võiksid veel olla ja nõuded, mida ei ole esialgu vaja [7].

Kategooriad on määratletud järgmiselt:

- „Peavad olema“ nõuded on kasutusele võtmise jaoks üliolulised, et need oleksid edukad [7].
- „Peaksid olema“ nõuded, on olulised, kuid pole kasutusele võtmise selles ajaraamis vajalikud [7].
- „Võiksid olla“ nõuded pole vajalikud, kuid võivad parandada arenduskulude eest kasutajakogemust või klientide rahulolu. Need lisatakse tavaliselt siis, kui aeg või eelarve seda võimaldab [7].
- Sidusrühmad lepivad vähemalt kriitiliste punktidenä kokku, et nõuded, mis on märgistatud kui „mida ei peaks täitma“, on vähemalt olemas ja kaardistatud. See ei tähenda, et need poleks väärtuslikud – lihtsalt neid pole hetkel vaja ja neist saab kas loobuda või lükata hilisemasse arendusse [7].

Käesoleva töö raames lahendatakse ära neist esimene ja osa teisest prioriteetsusest ning ülejäänud jäävad hilisemateks arendusteks.

Kohustuslikud nõuded:

- Isiku profiili loomine, muutmine ja salvestamine andmebaasi nii ametnikele kui ka külastajatele.
- Kasutajarollide loomine ja muutmine vastavalt kasutajate õiguste vajadusele. Rollid võivad olla nii vanglapõhised kui ka ülesüsteemilised.
- Isiku profiilil peab asuma info, kas talle on lubatud siseneda vangla territooriumile tehnikavahendi või sõidukiga.
- Isiku profiilil peab asuma info, kas ta viibib antud hetkel vangla territooriumil.
- Sõidukile luuakse profiil, mida on võimalik muuta ja salvestada ning millele on võimalik pädeva ametniku poolt anda lubasid koos juhiga sisenemiseks.
- Sõiduki profiilil peab asuma info, kas ta viibib antud hetkel vangla territooriumil. Lisaks peab profiilil asuma info kõikide juhtide kohta, kellele vangla on andnud loa sõidukiga territooriumile sisenemiseks.
- Juhul, kui puudub võimalus menetleda sissepääsulubasid käesolevas infosüsteemis, peab olema võimalik sisestada andmeid antud lubade kohta.
- Tehnikavahendile luuakse profiil, mida on võimalik muuta ja salvestada andmebaasi. Tehnikavahend seotakse alati mingi isiku või asutusega.
- Ühe loa võib olla seotud mitu sisenejat, sõidukit ja tehnikavahendit. Samas, loa andjaid on üks. See tähendab, et loa andja saab anda igale loal olevale isikule ja temaga kaasas olevale tehnikavahendile või sõidukile loa sisenemiseks. Samamoodi saab ta keelduda nt tehnikaloo lubamisest territooriumile, kuid isikule võib ta loa siiski anda.
- Peab olema võimalik sisestada erineval hulgal andmeid sõltuvalt loa liigist. Õigusaktidest tuleneva loa puhul on sisestavate andmete hulk suurem kui näiteks

alalise või lühiajalise sissepääsu loa korral. Täpsem andmete hulk on loetletud olemi-suhte diagrammis.

- Väljastatud lubasid peab olema võimalik tunnistada kehtetuks enne loal antud tähtaja saabumist. Kehtetuks tunnistamine peab toimuma reeglina sama ametniku poolt (ametikohta järgi), kes loa andis.
- Isikuid, sõidukeid ja tehnikavahendeid peab olema võimalik pääsلاس sisse ja välja registreerida selliselt, et tekiks ülevaade nii isikutest, sõidukitest kui ka tehnikavahenditest, mis on mingil ajahetkel vangla territooriumil. Vangid ei ole sihtrühm.

Soovituslikud nõuded:

- Ajalugu ja logimine. Ajalugu tähendab, et süsteem võimaldab pärida andmeid, mis olid päringu tegemise kuupäeva seisuga (hilisemad muutmised arvesse ei lähe). Kasutaja tegevused logitakse. Eeldab ka nn *Soft-Delete* ja *Soft-Update* kasutamist.
- Isiku profiilile kuvatakse märke, kui ta kuulub inimeste hulka, keda läbi ei otsita, välja arvatud juhul, kui selleks on justiitsministeeriumi ametniku luba.
- Loodavas infosüsteemis peaks toimuma ka vanglasse sisenemise lubade menetlus. Ehk et protsess peaks toimuma kellegi soovist kuni otsuseni. Protsessi jooksul peavad erinevad osapooled saama lubasid kooskõlastada või jätta kooskõlastamata koos põhjenduse märkimise võimalusega.
- Sisenemist võimaldavate lubade puhul peab olema võimalik paindlikult kehtestada sisenemise ajavahemike reegleid sarnaselt MS Outlook kalendriga (ühekordsed või korduvad sõltuvalt nädalapäevast, tööpäevast jne).

Nõuded, mida võiks täita:

- Vangide andmed päritakse KIR'st, vangidele profiili ei looda.
- Peab olema võimalik nii isikute kui ka sõiduki vanglasse sisenemise õigust peatada.

- Ametnike kasutajahalduse paremaks toimimiseks tuleks loodav infosüsteem liidestada personalihaldustarkvaraga. Selle abil võiksid nii kasutajaõigused kui ka vangla(te)sse sisenemise õigused toimuda automaatselt vastavalt andmevahetusele. Justitiisüsteemis kasutatakse personaliandmete haldamiseks SAP'i.
- Loa andja või kooskõlastaja(d) sõltub(vad) loa liigist.
- Sisenemiste kohta peab olema võimalik sisestada ka sündmusi vms, kust oleks võimalik saada hiljem infot selle kohta, kui isik on nt unustanud või võtnud sihilikult kaasa vanglas keelatud eseme. Sisestatud sündmusi peab olema võimalik hiljem muuta ja vajadusel kustutada.
- Aruannete funktsionaalsus. Peab olema võimalik koostada erinevatest otsinguparameetritest lähtuvaid aruandeid andmebaasist. Aruanded on samuti kasutusõigustega piiratud.
- Päringu andmeid peab olema võimalik importida tabelitöötlusprogrammi.

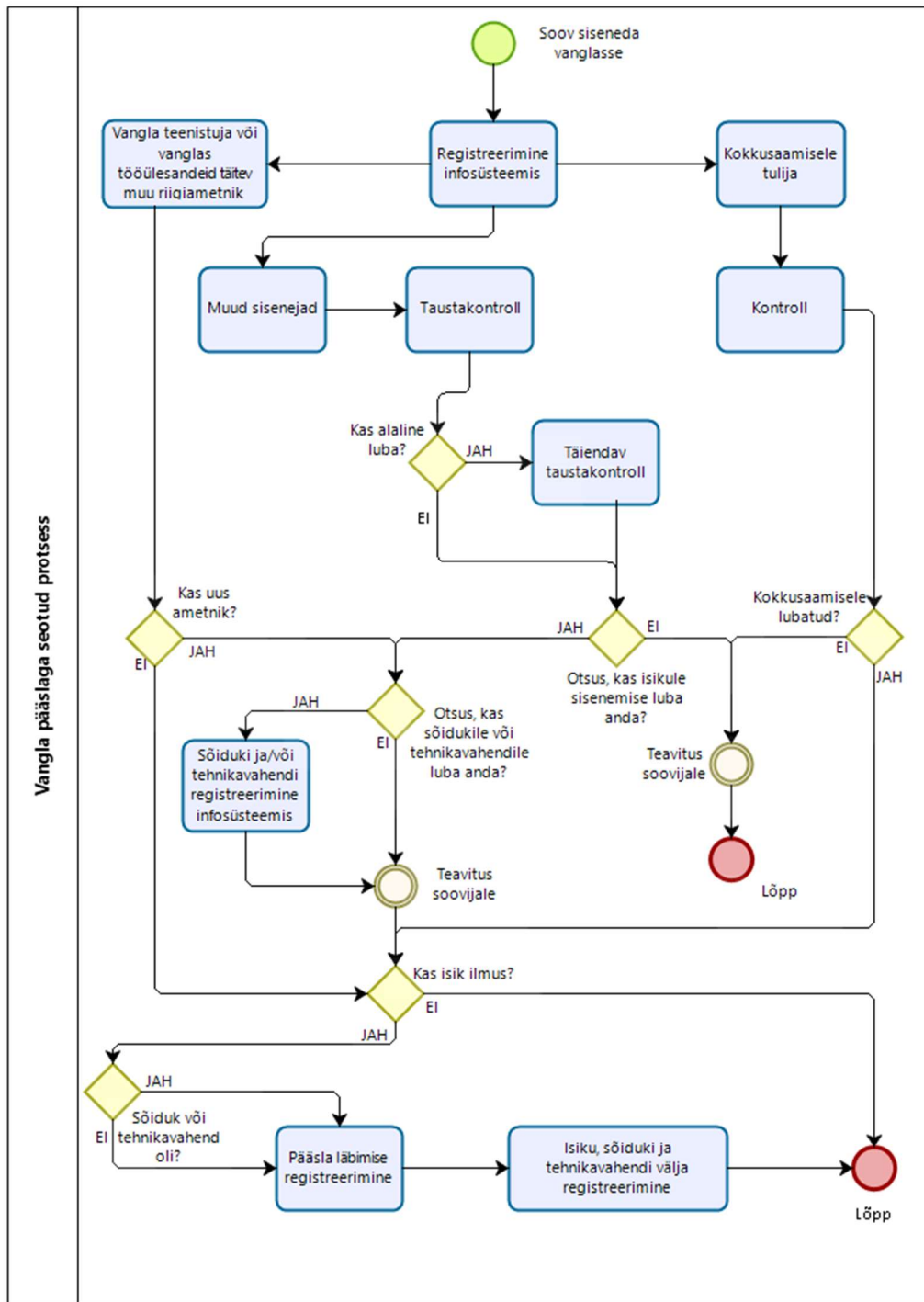
Nõuded, mida esialgu ei peaks täitma:

- Isiku profiili koostamisel kasutatakse riistvara, mis loeb isiku identifitseerimist võimaldavalt dokumendilt vajalikud (olemi-suhte diagrammil loetletud) metaandmed loodavasse infosüsteemi.
- Isiku profiilile salvestatakse tema pilt. Vajadusel tehakse pilt vanglas, mitte ei loeta isikult tõendavalt dokumendilt.
- Isiku profiili peab olema võimalik siduda ka vanglas kasutatavate elektrooniliste uksekaartidega. Neid seoseid peab olema võimalik vaadata, lisada, muuta ja kustutada.
- Käesolev süsteem saadab välja vastused/teavitused lubade taotlustele – nii lubavad kui ka keelduvad. Teavitused lähevad välja adressaadile ka loa kehtetuks tunnistamise korral.

- Isikute sisenemist peab saama registreerida ka grupina – nt vanglate relvastatud üksuse sisenemise korral. Neid andmeid peab olema võimalik muuta ja täiendada ka grupi vanglast välja liikumise järgselt.
- Sõidukite sisenemine toimub numbrituvastamise kaamera abil nii, et sõiduk sõidab vangla transpordilüüsi ukse juurde, kaamera tuvastab sõiduki, annab sellest pääsla ametnikule teada, mille abil pääsla ametnik näeb juba ette ära sõiduki ja juhiga seotud andmed ja neile väljastatud loa andmed. Sellisel juhul jääb vanglaametnikul üle vaid veenduda, et vanglasse sisenev või sealt väljuv sõiduk ja isik ning tehnikavahendid on samas koosseisus loale märgituga.
- Käesoleva süsteemiga ühendatud kaamera abil peab olema võimalik tuvastada, kas skaneeritud dokumendil olev isik on sama isik, kes vanglasse siseneb või sealt väljub.
- Külastajale prinditakse välja külastajakaart koos pääsلاس isikust tehtud pildiga.
- Süsteemi autentimine peaks toimuma ID-kaardi, mobiil-ID ja AD-konto abil.
- Liidestamine vangide andmeid koondava infosüsteemiga selleks, et võtta vastu ja saata andmeid, mis on seotud vangide või nende lähedaste pääsلاس läbimisega.

3.6 Äriprotsessi lihtsustatud skeem

Käesolevas alampeatükis näitab autor, millisena võiks toimuda äriprotsess pääsلاس infosüsteemi kasutusele võtmisele korral. Tegemist on süsteemi minimaalsete nõuete järgi koostatud skeemiga.

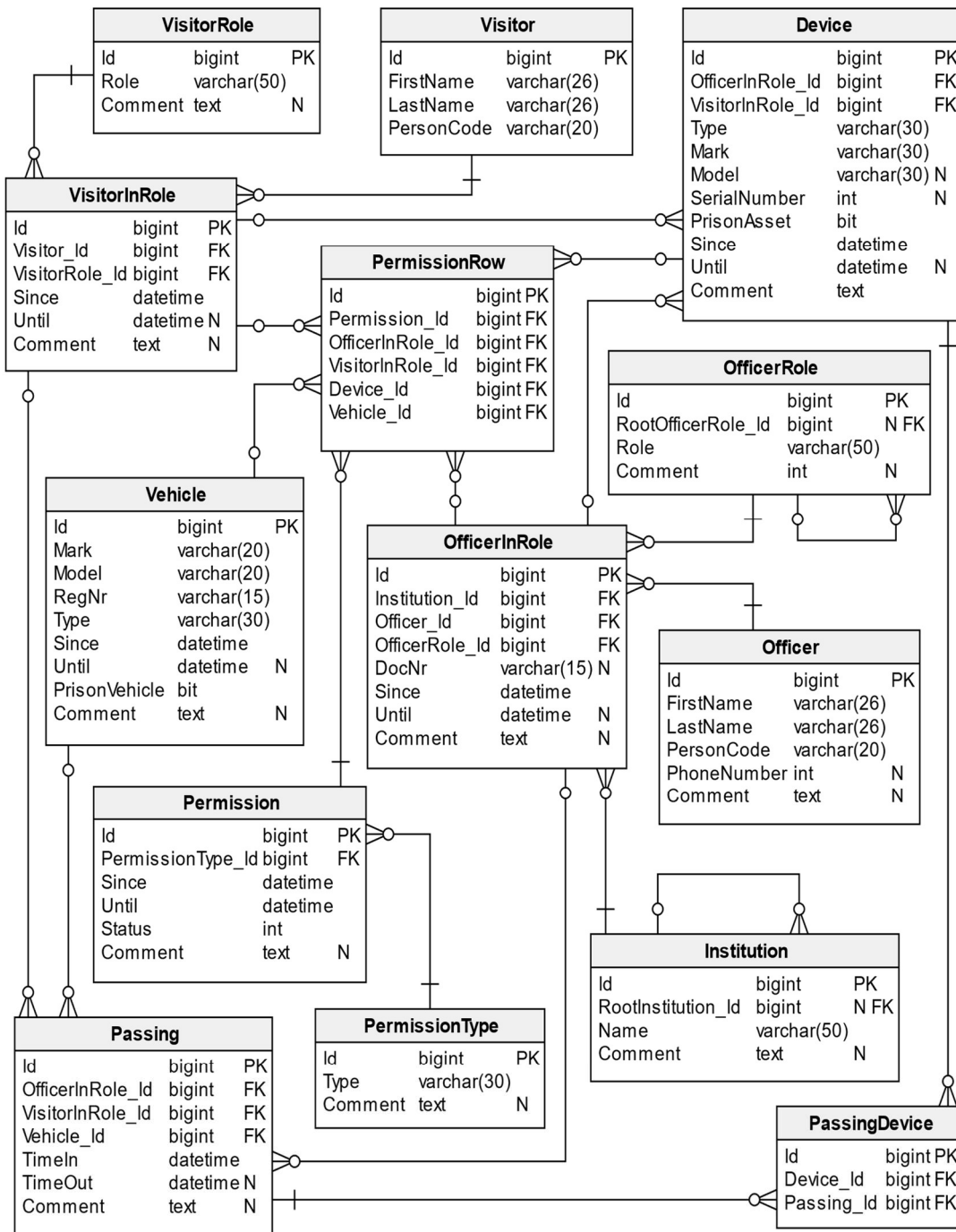


Joonis 1. Äriprotsessi lihtsustatud skeem.

3.7 Olemi-suhte diagramm

Kui nõuded on kogutud ja analüüsitud, on järgmine samm luua kontseptuaalne skeem andmebaasi jaoks, kasutades kõrgetasemelist kontseptuaalset andmemudelit [2]. Sellest järgmine samm andmebaasi kujundamisel on andmebaasi tegelik juurutamine, kasutades DBMS-i. Seda sammu nimetatakse loogiliseks kujunduseks või andmemudeli kaardistamiseks ning selle tulemuseks on andmebaasi skeem DBMS-i andmemudeli rakendamises [2]. Kuna käesoleva töö eesmärk on rakenduse arendamine, siis jäetakse nende sammude kirjeldamine käesoleva töö ulatusest välja ning tuuakse vaid tulemus.

Käesolevas alampeatükis esitab autor olemi-suhte diagrammi, mis on koostatud lähtuvalt käesolevas peatükis toodud ärilistest nõuetest.



Joonis 2. Pääsla infosüsteemi olemi-suhte diagramm.

4 Arendus

Käesolevas peatükis analüüsitakse töö esemeks oleva infosüsteemiga seotud tehnilisi valikuid.

4.1 RIK-iga seotud tehnilised nõuded

RIK'il kehtis kuni 2020. aasta veebruarini direktori käskkirjaga kinnitatud IT-profiil [8]. Vaatamata sellele, et tegemist ei ole enam kehtiva asutuse juhi käskkirjana, ei tähenda see absoluutselt seda, et seal väljatoodud printsiibid oleksid praktikas muutunud. Pigem on need põhimõtted liikunud avalikust allikast kohati piiratud käideldavusega hangete tehnilistesse tingimustesse.

RIK'i ülesanded on toodud tema põhimääruses, mille kohaselt tagab RIK justiitsministeeriumi (tegelikkuses ka aadressil Suur-Ameerika 1 asuvate teiste ministeeriumite) ja ministeeriumi valitsemisalas olevate asutuste ning vastavalt kokkulepetele ka teenindatavate asutuste info- ja sidesüsteemide töökindluse ning nõuetele vastava kasutatavuse ja arengusuutlikkuse [9].

Seni on olnud RIK'i poliitika selline, et haldusala töötajate kasutatavateks seadmete operatsioonisüsteemide tarkvaraks on olnud Microsoft'i toodetud Windows. Üksnes väga üksikutele kasutajatele on võimaldatud erandeid: kes paiknevad võtmekohtadel ja kelle tungivaks nõudmiseks on olnud Apple riistvara ning nende seadmetel kasutatav tarkvara MacOS. Samas on RIK üritanud mõned aastad tagasi minna üle ka vabavaralisele tarkvarale (katsetati esialgu vabavaralist kontoritarkvara), kuid see ei õnnestunud kasutajate vastuseisu ja teatud juhtudel tarkvara ühildamatuse tõttu. Seetõttu tuleb RIK'i infotehnoloogiliste arenduste tegemisel lähtuda nõudest, et süsteemid peavad töötama operatsioonisüsteemiga Windows ja ühilduma Microsofti kontoritarkvaraga.

Tarkvara standardkonfiguratsiooni poole pealt on olnud varasemalt nõudeks, et tarkvara programmeerimiskeelteks peavad olema C#, Java või Python. [8] Kuigi RIK'i direktori käskkiri on kehtetu, ei ole RIK'i portfell muutunud. Üle 75% arendatud ja RIK-i vastutada

olevast tarkvarast on arendatud programmeerimiskeeles C#. Vanglate valdkonnas suurim infosüsteem KIR on arendatud just keeles C# ja seda keelt kasutavas tarkvararaamistikus .NET ning selle alla kuuluvast veebiraamistikus ASP.NET.

Andmebaasitarkvara tootjate puhul olulisi piiranguid ei ole ja kõik reaalselt suurte tarkvaralahenduste poolt kasutatavad andmebaasisüsteemid on RIK'i poolt välja toodud, seal hulgas vabavaraline lahendus. Nendeks on Oracle, Microsoft SQL ja Postgre SQL [8].

4.2 Arenduskeele valik

Maailmas on ligikaudu 700 erinevat programmeerimise keelt [10]. Enamus neist ei oma erilist populaarsust programmeerijate seas. Populaarsuse edetabeleid koostatakse palju erinevaid lähtuvalt sellest, mida edetabeli koostaja oluliseks peab. Üldiselt domineerivad erinevates edetabelites Python, Java, C, C++, C# ja JavaScript.

Rakenduse keele valikul tuleb lähtuda sellest, et arendatav infosüsteem peab sobima RIK'i ja vanglate valdkonna arendajate profiiliga, milleks tänasel päeval olulisimaks on programmeerimiskeel C#.

Kui jätta eelmises punktis ja ka käesoleva punkti eelmises lõigus toodud nõuded kõrvale, tuleb siiski asuda seisukohale, et C# oleks ka ilma nendeta autori eelistatuim keel RIK-i kolme põhikeele hulgast. Kui võrrelda omavahel Javat ja C#, siis viimane on oluliselt paindlikum, näiteks ei ole vaja keeles C# teha igale avalikule klassile luua eraldi faili, nagu seda nõuab Java. Ilma selliste piiranguteta on projekt paremini hallatav. Lisaks on keeles C# oluliselt paremini lahendatud geneeriliste tüüpide ja refleksiooni tugi. Javas on geneerilised andmetüübid kaduvad andmetüübid, mis tähendab, tüübi info kaob kompileerimise järgselt ära. Seevastu keeles C# geneeriliste andmetüüpide tüübi info säilib ka pärast kompileerimist ja seega on see info kättesaadav ka refleksiooni vahenditega programmi töö ajal. Tõenäoliselt kõige olulisem faktor C# kasuks peitub rakenduse ja andmebaasi vahelises suhtluses. Kui Javas andmebaasi päringute koostamine sõltub selle kasutatavast raamistikust, siis C#'s on keelde sisse ehitatud väga võimas tööriist nimega LINQ. LINQ loodi üle 10 aasta tagasi Microsofti poolt. Selle näol on tegemist tehnoloogiaga, mis pakub programmeerimiskeele tasemel tuge igat tüüpi andmete pärimiseks. LINQ võimaldab teisendada koodist andmeid objektideks, XML'i,

SQL'i, olemiteks, andmehulkadeks ja vastupidi [11]. Kuigi Java arenduskeelde tekkis 2014. a. Stream API, on LINQ kasutamine oluliselt mugavam.

Pythonit ja C# on omavahel keerulisem võrrelda, kuna mõlemad on oma raamistikega võimsad ja paindlikud. Pythoni miinuseks on peetud tema aeglust, kuid tänaste arvutite võimsuse juures ei ole see enam tõsiselt võetav faktor, kuna sellise veebirakenduse puhul mõjutavad kiirust enam suhtlus andmebaasiga. Algaja programmeerija jaoks on kindlasti olulisel kohal kompileerimisaegne staatiline analüüs, mida Pythonis ei ole, kuna see on pigem interpreteeritav keel, kuid C#'s on. Pythoni eeliseks saab lugeda arvukaid teke, mis teevad koodi kirjutamise kergeks ja kiireks, kuid kindlasti ei jää C# selles palju maha. Autori jaoks on C#'l eeliseid siiski rohkem ja seetõttu oleks arenduskeele valik langenud sõltumata RIK-i profiilist C# kasuks.

4.3 Rakenduse platvormi valik

Autor eeldab, et käesolevas bakalaureuse töös ei ole tarvis lahti seletada veebirakenduse ja töölauarakenduse tehnoloogilisi erinevusi, vaid piisab sellest, kui käsitleda nende tehnoloogiate eeliseid ja puudujääke selleks, et põhjendada autori valikuid tehnoloogia valimisel.

Järgnevalt toob autor välja nii veebirakenduse kui ka töölaua rakenduse eelised ja puudused lähtudes viidatud allikas käsitletud informatsioonist [12].

Veebirakenduse eelised:

- Ei vaja installeerimist. Rakendus on ligipääsetav erinevatest asukohtadest sõltumata sellest, kas kasutatavasse seadmesse on rakendus installeeritud.
- Automaatsed tarkvara uuendused toimuvad ilma kasutaja sekkumiseta, kui kasutaja on ennast rakendusest välja loginud.
- Platvormist sõltumatu ligipääs. Veebirakendusega on võimalik ühenduda suvalisest operatsioonisüsteemist, milles on võimalik kasutada rakenduse toetatud veebilehe sirvijat.
- Mobiilne ligipääs. Enamik veebirakendusi saab töötada ka mobiilseadmetes. Sõltuvalt programmi olemusest ja funktsionaalsusest võib see mõnikord olla

piiratud ekraani suuruse või vähem täpse navigeerimise tõttu. Kuid valdav osa ettevõtte veebilahendustest toimivad süsteemist sõltumata täiesti hästi.

- Tarvitatavad ressursid. Veebiteenused kasutavad oluliselt vähem protsessori töötlemisvõimsust.

Veebirakenduse puudused:

- Turvalisusega seotud nüansid. Alati, kui mõni teave asub kusagil kaugserveris, tekib turvalisuse küsimus. Kuid see ei kehti üksnes veebiteenuste puhul. Samas on veebiserverid potentsiaalsete ründajate eelistatav sihtmärk.
- Sõltuvus internetiühendusest. Pole kahtlust, et juurdepääs internetile pole kunagi varem olnud usaldusväärsem, kuid pole välistatud sellega seotud tõrked. Paljudel kaasaegsetel tarkvaralahendustel on selle vastu kaitsemeetmed. Kui need on brauserisse laaditud, saavad nad ühenduse katkestamisel kasutada kliendipoolset mälu. See võimaldab jätkata tööd ja värskendada serveripoolseid andmebaase automaatselt kõikjal, kus ühendus taastatakse.
- Aeglasem kui tööluarakendus. Kuna veebirakendustel pole juurdepääsu nii suurele töötlemisvõimsusele, on rakendused loomulikult töölaua ekvivalentidega võrreldes aeglasemad.

Tööluarakenduse eelised:

- Interneti-ühendus pole vajalik. Nii laialt levinud kui see ka pole, pole internet veel saavutanud 100% usaldusväärset. See sobib lauaarvutite jaoks. Kui rakendus on arvutis, saab seda kasutada ilma ühenduse viivitusega, kuigi andmeid ilma internetiühenduseta uuendada pole võimalik.
- Privaatsuse küsimused. Kui võrd teenus toimib sisevõrgus VPN-i kasutades, ei ole ohud privaatsusküsimuste riiveks tõsiselt võetavad.
- Parem jõudlus. Veebirakendustega võrreldes on tööluarakendustel kahtlemata parem jõudlus. Samad toimingud viiakse lõpule kiiremini ja funktsioonide valik on sageli suurem.

- Arvutiressursside optimaalne kasutamine. Kasutades arvuti töötlemisvõimsust, saavad töölauarakendused täita paljusid funktsioone, mis pole veebirakenduse jaoks teostatavad. Näiteks kolmemõõtmelise graafika töötlemine.

Töölauarakenduse puudused:

- Pole kaasaskantav. Nagu nimigi ütleb, on töölauarakendused fikseeritud töölauale. Kuid see on endiselt fikseeritud ühe süsteemiga. Ei saa lihtsalt kellegi teise arvutisse sisse logida ja jätkata rakendustega töötamist.
- Kohustuslik paigaldamine. Iga töölauarakendus tuleb arvutisse paigaldada.
- Kõvakettaruum. Nii programm ise kui ka selle loodud failid salvestatakse kõvakettale. Mõni rakendus ei pruugi palju ruumi kasutada, kuid video- ja graafikatöötlemine võivad võtta palju ketta ruumi.
- Käsitsi värskendused. Veebirakendused on alati ajakohased. Uusim versioon edastatakse kasutajatele automaatselt, kui rakendus käivitatakse. See ei kehti töölaua tarkvara kohta. See võib teile teatada, et värskendus on saadaval, kuid peate selle siiski käsitsi alla laadima ja paigaldama.

Majutamise kulude katmine ei ole käesoleva infosüsteemi puhul küsimuseks, kuna see majutatakse domeenis JUSTMIN. Teenuse lisandumisega ei kaasne olulisi kulusid.

Eelpool toodud plussid ja miinused annavad käesoleva töö olemuse tõttu selge eelistuse veebirakenduse kasuks. Turvalisusega seotud riskid on maandatud läbi selle, et arendatav infosüsteem, nagu justiitsministeeriumi teised infosüsteemid, saab olema ligipääsetav vaid justiitsministeeriumi sisevõrgust või välisvõrgust üksnes juhul, kui kasutatakse tööandja arvutit ja arvuti on ühendatud keskseadmetega kasutades VPN-i.

Jõudlus ei ole riikliku andmekogu puhul reeglina oluline murekoht, kuna tavapäraselt liiguvad serveri ja kasutaja arvuti vahel vähe mahtu nõudvad andmed ning erinevad ühendused on kiired.

Samas on mobiilsus kasutajate jaoks kõige olulisem nõue – infosüsteem peab olema ligipääsetav suvalisest justiitsministeeriumi arvutist ilma, et rakendus oleks eelnevalt seadmesse paigaldatud.

4.4 Raamistike valik

4.4.1 Üldine raamistik

Niipalju kui informatsiooni on kätte saada, võib väita, et C# ja .NET on omavahel väga tihedalt seotud, mis tähendab, et C# ilma .NET-ta ei ole mõistlik kasutada. Seetõttu on ainuvõimalik kasutada C# valikul just .NET raamistikku. Lisaks on .NET Core mõeldud kasutamiseks enimlevinud platvormidel, mis teeb sellest platvormist sõltumatu raamistiku.

Kuna .NET Core on avatud lähtekoodiga ja kogukonna poolt arendatav raamistik, siis seda täiendatakse pidevalt ja uueneb sageli. See teeb .NET raamistikust hea valiku pikaajase tarkvara arendamisel.

4.4.2 Veebiraamistik

Põhjused, et käesolevas töös arendatakse veebirakendust, siis tuleb kasutada ka raamistikku ASP.NET Core, mis on üks osa raamistikust .NET ja mõeldud just veebirakenduste arendamiseks. ASP.NET Core muudab veebirakenduste kirjutamise kiiremaks, lihtsamaks ja turvalisemaks. See sisaldab teke üldiste teemade jaoks:

- Dünaamiliselt muutuvate veebilehtede loomine
- Kasutajate autentimine veebirakendusse
- Võimaldab kasutajatele OAuthi abil veebirakendusse autentimiseks kasutada oma Google'i, Facebooki või muud kontot
- Üldise struktuuri pakkumine rakenduste loomiseks
- Konfiguratsioonifailide lugemine
- Piltide failide serverimine
- Veebirakenduses kasutajate tegevuse logimine [13]

4.4.3 ORM raamistik

Kuna käesolevas töös on kasutusel C# ja .NET, siis on ainumõistlik kasutada ORM raamistikuna EF Core'i. EF Core kasutab .NET'i raamistiku LINQ'i soovitud käskude edastamiseks ja tavalisi .NET'i klasse andmete hoidmiseks. LINQ-i päring pääseb ligi rakenduse *DbContext*'i atribuudile ja laeb baasist välja ka selle atribuudiga seotud suhted. Selle teisendab andmebaasipakkuja SQL-i käsuks, et teostada päringuid andmebaasi. Saadud päringu vastus salvestatakse vahemällu, et vältida kulukat andmebaasi päringu kordamist. EF Core püüab andmebaasipääsudel olla võimalikult tõhus. Mitmest olemist andmeid küsides ühendab LINQ need ühe päringuga üheks suureks tabeliks. Pärast seda, kui andmebaasi pakkuja on andmed läbi lugenud, loob EF Core .NET-klasside objektid ja kasutab andmebaasi välisvõtmeid .NET-klasside õigeks linkimiseks. Tulemuseks on .NET klassi objektide kogum, mis on kõik omavahelises õiges seoses [14]. Alternatiiv EF Core'ile oleks nt Dapper, mis oma kiirusega ületab kõiki konkurente. Kuid selle miinuseks on selgelt kasutusmugavus. Näiteks ei saa sellega kasutada LINQ'i, mis tähendab, et kogu andmebaasiga suhtlus tuleks endal SQL'is kirjutada.

4.5 API disain

Kuigi käesoleva töö ulatus puudutab vaid tagarakendust, siis käsitleb autor käesolevas punktis laiemalt lahendust, kuidas kasutaja saaks tulevikus süsteemi kasutada. Kuna käesolevas projektis kasutatakse raamistikuna ASP.NET'i, siis on valikud rakenduse kasutajaliidesega suhtluseks järgmised:

- ASP.NET poolt eksklusiivselt pakutav RazorPage;
- MVC disain;
- REST API;
- SOAP API.

Kahe esimese väljatoodud valiku puhul pakub rakendus ise kasutajaliidest võimalikele kasutajatele. Kahe viimase valiku puhul tuleb kasutajaliidesed eraldi esirakendusena programmeerida. Kahe esimese valiku puhul on selgeks miinuseks lehtede funktsionaalsus. Nimelt ei ole võimalik nende abil kasutada erinevaid dünaamilisi

atribuute ilma, et vahepeal toimuks andmevahetust esi- ja tagarakenduse vahel. Igasugune andmevahetus teeb rakenduse kasutamise oluliselt aeglasemaks. Seega on esimesed kaks lahendust oma funktsionaalses mõttes piiratumad ja need ei sobi võimalikele kasutajatele. Turvalisuse, ennekõike autentimise, osas suuri erinevusi ei ole – kõik sõltub programmeerija valmidusest see osa hästi programmeerida. Esimesel kahel juhul teeb suure osa programmeerija eest ära ASP.NET, teisel juhul tuleb rohkem tööd teha programmeerijal.

Viimase kahe puhul on oluline märkida, et SOAP API arendus on oluliselt keerulisem kui REST API arendus. Samas saadav kasu ei ole suurem kui REST API puhul. Mõlemad tehnoloogiad võimaldavad kasutada kõiki dünaamilisi atribuute, mida tänapäevane JavaScript võimaldab. Samas kaasneb SOAP puhul arendajale rohkem tööd nii taga- kui ka esirakenduse programmeerimisel. Lisaks on SOAP andmevahetuseks kasutatava mahuka XML formaadi tõttu aeglasem kui REST tehnoloogia ja seetõttu kulub selle sõnumi sisu koostamisele ja lugemisele rohkem aega. Seda ilmestavad järgnevad joonised 3 ja 4.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <soapenv:Envelope
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xmlns:api="http://127.0.0.1/Integrics/Enswitch/API"
5   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
6   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope">
7   <soapenv:Body>
8     <api:some_api_call soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding">
9       <username xsi:type="xsd:string">foo</username>
10      <password xsi:type="xsd:string">bar</password>
11    </api:some_api_call>
12  </soapenv:Body>
13 </soapenv:Envelope>
```

Joonis 3. Kahe andmeväljaga SOAP päringu näidis

REST päringu puhul tuleb teha SOAP tehnoloogiaga võrdluseks allolev päring URL-ile.

```
1 {
2   "username": "foo",
3   "password": "bar"
4 }
```

Joonis 4. Kahe andmeväljaga REST päringu näidis

4.6 Autentimine

Üldiselt on olemas kolm erinevat tunnustatud autentimise liiki, mis on liigitatud nende olemuse järgi:

1. Autentimine millegagi, mida kasutaja teab. Siia alla kuuluvad näiteks paroolid, PIN-koodid jms.
2. Autentimine millegagi, mis on kasutajal olemas. Siia alla kuuluvad erinevad füüsilised seadmed, näiteks kiipkaardid.
3. Autentimine kasutaja endaga. Siia alla kuuluvad ennekõike biomeetrilised andmed nagu näiteks näotuvastus, sõrmejalg, silmaiiris [15].

Tavapäraselt soovitatakse turvalisuse tõstmise eesmärgil autentimist kahe või enamat liiki autentimise abil. Ehk näiteks kasutaja peab autentimisel lisaks paroolile omama ka kindlat kiipkaarti. Eestis on sellisteks abivahenditeks loodud autentimise ja digitaalse allkirja andmise vahendid ID-kaart, mobiil-ID ja Smart-ID. Nende puhul on vajalik mingisugune füüsiline seade ja parooli või koodi teadmine.

ASP.NET puhul on autentimise halduseks mitmeid erinevaid lahendusi (lisaks pakub ASP.NET kahefaktorilist autentimise lahendust [16]), kuid enamjaolt on nad seotud üksnes paroolide kasutamisega. Ehk eeltoodud liigituse järgi lähevad nad esimese alla.

Riigiasutuste puhul on pigem nõudeks, mitte soovituseliseks, see, et kõiki andmeid (ennekõike asutusesiseseks kasutamiseks ettenähtud andmeid) hoitakse asutuse või seda teenindava IT-asutuse serverites. Seetõttu ei ole mõeldav kasutada ametnike autentimiseks nt avalike eraettevõtjate OAuth teenuseid.¹ Selle standardi järgi ise

¹ OAuth on juurdepääsuvolituste avatud standard, mida tavaliselt kasutatakse Interneti-kasutajate veebisaitidele või rakendustele juurdepääsu andmiseks, parooli andmata [17].

rakenduse autentimise arendamine ei ole mõistlik. Lisaks ei paku piisavat turvalisust ka muud ASP.NET'i kasutamise seotud autentimise lahendused, kuna neis kasutatakse siiski vaid üht liiki autentimise lahendust. RIK-il on kasutusel enda autentimise portaal, mille abil on võimalik ametnike rakendustesse autentida. Autentimiseks saab kasutada AD konto andmeid, ID-kaarti või mobiil-ID lahendust. Kuigi parooli kasutamine on seotud vaid ühe liigi autentimise liigi kasutamisega, saab selle lahenduse ametnike jaoks kinni panna, kui kõigile süsteemi kasutavad ametnikud on teadlikud piirangutest süsteemi kasutamisel ja on endale hankinud kas ID-kaardi või mobiil-ID, mida igapäevaselt tööalaselt kasutada.

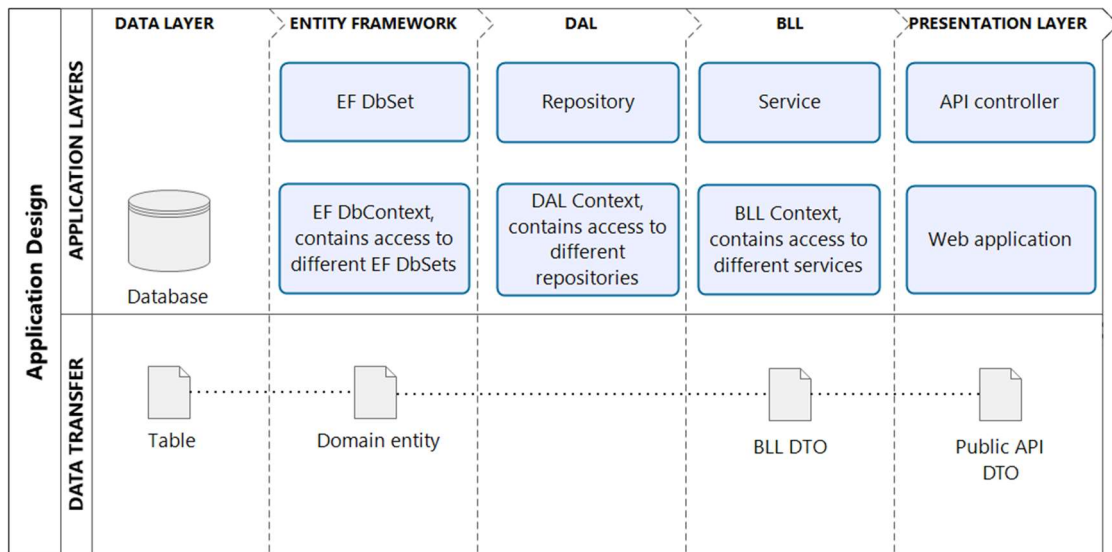
Kuna RIK'i autentimisportali kasutamine ei nõua olulisi lisaarendusi ja tegemist on väga turvalise lahendusega, siis on parim võimalik lahendus just selle kasutamine.

4.7 Rakenduse arhitektuur

Käesolevas jaotises on esitatud arendatava infosüsteemi rakenduse üldine arhitektuur. Rakendus koosneb erinevatest kihtidest:

- Andmebaas – andmekiht
- Andmebaasiga suhtlemiseks vajalik kiht, mis on rajatud EF'i abil
- Äriloogika kiht
- Kasutajale esitluseks vajalik kiht

Kihtide vaheliseks andmete liigutamiseks on koostatud andmevahetuseks vajalikud klassid (*DTO*).



Joonis 5. Rakenduse arhitektuuriline skeem.

4.8 Dokumentatsioon

Rakenduse arenduses kirjutatud programmeerimise koodi dokumenteerimiseks on saadaval väga palju häid lahendusi. Nende omavaheline võrdlemine ei anna palju juurde, vaid tuleb välja valida see, mida kasutada. Kuivõrd koolis õpetati vaid üht, siis kasutatakse ka käesolevas lõputöös just seda tehnoloogiat dokumentatsiooni raamistiku loomiseks. Nimelt on REST API dokumentatsiooni koostamise jaoks kasutatav tehnoloogia *Swagger*¹.

¹ Swaggeri kasutajaliides võimaldab API ressursse visualiseerida ja nendega suhelda lihtsalt OpenAPI spetsifikatsiooni järgi [18].

5 Töö tulem

Töö tulemusena valmis vangla pääsla infosüsteemi REST API disainil põhinev veebirakenduse tagarakendus veidi enam kui minimaalses vajalikus ulatuses. Selle abil on võimalik vanglas kaotada ära tänased tavalises kontoritarkvaras peetavad tabelid ja võtta pärast esirakenduse arendamist kasutusele kõigi kolme vangla peale ühine pääsla infosüsteem.

Töö skoop katab ära nii lubade, inimeste, sõidukite kui ka inimestega kaasas olevate tehnikavahendite registreerimise, lubade alusel vanglasse sisenemise ja väljumise registreerimise ning sissepääsulubade haldamise funktsionaalsuse. Lisaks on võimalik kõiki andmeid lugeda, muuta ja kustutada. Tagarakendus kontrollib ka isikukoodi korrektsust ning isikukoodi alusel topelt isikute sisestamise soovi. Valminud arendus võimaldab ka päringu vastuse tulemusi sorteerida ja filtreerida.

Välja toodud funktsionaalsus on juba enam kui piisav võrdluses tänase lahendusega. Enne infosüsteemi kasutusele võtmist on tarvis arendada sellele ka esirakendus, mille kaudu kasutaja saaks tagarakendust ja andmebaasi kasutada.

Edasiarendamise vajadused on toodud käesoleva töö analüüsi kolmandas peatükis. Pikas perspektiivis tuleks infosüsteemi laiendada nii inimeste isikut tõendavate dokumentide metaandmete pääsla infosüsteemi lugemisega kui ka dokumendifoto ja reaalse inimese tarkvaralise võrdlusega. Väiksematest edasi arendamise vajadustest väärivad välja toomist andmevahetus vangide andmeid koondava infosüsteemiga (vangidega seotud pääsla läbimised), andmete importimise tabelitöötlusprogrammi ja e-kirja kaudu liikuvad teavitused kasutajatele.

Siiski ei ole käesoleva projektiga plaanis asendada pääslas olevat inimest arvutisüsteemide vastu, vaid anda vanglale juurde üks lisaturvalisust tagav tööriist.

6 Kokkuvõte

Käesoleva bakalaureuse töö eesmärgiks oli arendada valmis vanglate pääsla infosüsteemi tagarakendus osas, mis täidaks vajalikud minimaalsed funktsionaalsed nõuded vanglate pääsla töö korraldamisel. Töö eesmärk sai täidetud. Infosüsteemi esirakenduse arendamise järgselt oleks võimalik infosüsteemi kasutusele võtta ning loobuda MS Exceli tabelite kasutamisest sisenejate ja väljujate registreerimisel.

Töö teises peatükis kirjeldati ja selgitati probleemi olemust ning põhjendati valitud töö teema aktuaalsust. Töö kolmandas peatükis toodi välja õigusaktidest tulenevad tähtsamad õiguslikud nõuded ning seejärel MoSCoW mudeli järgi ärilised nõuded. Tähtsamad nõuded olid seotud olemasolevate Exceli tabelite väljavahetamiseks vajalike nõuetega: isikute ja sõidukite sisestamise võimalus, lubade haldus, isikute sisenemise ja väljumise registreerimine ning päring selle kohta, kui palju inimesi ja kes on mingil ajahetkel vangla territooriumil. Kolmandas peatükis toodi välja ka äriprotsessi kirjeldus skemaatiliselt ning andmebaasi ehitamise aluseks olev olemi-suhte diagramm. Peatükis in kirjeldatud ka edasised infosüsteemi arendamise vajadused.

Töö neljas peatükk keskendub infosüsteemi arendamisele. Selles peatükis selgitati ja põhjendati arenduse käigus tehtud valikuid: rakenduse platvormi valik, programmeerimiskeele valik, raamistike valik, API disain, dokumentatsiooni koostamiseks sobilik tehnoloogia ning autentimise lahendus. Selles peatükis jõuti järeldusele, et parimaks lahenduseks on API disainil põhinev veebirakendus, mille tagarakendus on arendatud programmeerimiskeeles C# ning mille arendamisel tuleb kasutada raamistikke .NET, ASP.NET ja Entity Framework.

Töö viies peatükk võtab kokku toot tulemused. Käesoleva töö väliselt jätkatakse esirakenduse arendamist selleks, et võtta infosüsteem vanglates kasutusele.

Kasutatud kirjandus

- [1] H. Liivak, Vangistuse täideviimine ja järelevalve korraldamine, Tallinn: Sisekaitseakadeemia, 2016.
- [2] S. B. N. Ramez Elmasri, Fundamentals of Database Systems (seventh edition), Pearson, 2016.
- [3] *Vangistusseadus*, 2000.
- [4] *Vabariigi valitsuse seadus*, 1995.
- [5] *Justiitsministri määrus "Järelevalve korraldus vanglas"*, 2011.
- [6] *Justiitsministri määrus "Vangla sisekorraeskiri"*, 2000.
- [7] H. PMO, „Hot PMO,“ <https://www.hotpmo.com/management-models/moscow-kano-prioritize/>, 2016.
- [8] RIK, „RIK,“ RIK, 2020. [Võrgumaterjal]. Available: <https://www.rik.ee/et/asutusest/it-profiil>. [Kasutatud 01 04 2021].
- [9] *Registrite ja Infosüsteemide Keskuse põhimäärus*, 2007.
- [10] E. autorid, „Wikipedia,“ 2021. [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/List_of_programming_languages. [Kasutatud 04 04 2021].
- [11] J. Joseph C. Rattz, Pro LINQ Language Integrated Query in C# 2008, Apress, 2007.
- [12] „Digital Skynet,“ 03 06 2020. [Võrgumaterjal]. Available: <https://digitalskynet.com/blog/Desktop-App-vs-Web-App-Comparative-Analysis>. [Kasutatud 02 04 2021].
- [13] A. Lock, ASP.NET Core In Action, Manning, 2018.
- [14] J. P. Smith, Entity Framework Core In Action, Manning, 2018.
- [15] J. M. Stewart, „Global Knowledge,“ 2018. [Võrgumaterjal]. Available: <https://www.globalknowledge.com/us-en/resources/resource-library/articles/the-three-types-of-multi-factor-authentication-mfa/>. [Kasutatud 17 04 2021].
- [16] Microsoft, „Multi-factor authentication in ASP.NET Core,“ Microsoft, 17 03 2020. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/aspnet/core/security/authentication/mfa?view=aspnetcore-5.0>. [Kasutatud 19 04 2021].
- [17] W. Gordon, „Understanding OAuth: What Happens When You Log Into a Site with Google, Twitter, or Facebook,“ lifehacker, 2012. [Võrgumaterjal]. Available: <https://lifehacker.com/understanding-oauth-what-happens-when-you-log-into-a-s-5918086>. [Kasutatud 18 04 2021].
- [18] S. KC, 2019. [Võrgumaterjal]. Available: <https://medium.com/@suraj.kc/use-swagger-to-generate-api-client-in-frontend-60b7d65abf31>. [Kasutatud 18 04 2021].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Priit Post

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Veebipõhise tagarakendi arendus pääsla läbipääsu infosüsteemi näitel“, mille juhendaja on Toomas Lepikult.
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

17.05.2021

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.