

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Karl Jensen 206204IADB

Kinnipeetavate apellatsiooniportaali prototüüp

Bakalaureusetöö

Juhendaja: Jaanus Pöial

PhD

Kaasjuhendaja: Mark-Hendrik Mäeste

Bakalaureusekraad

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: Karl Jensen

15.04.2023

Annotatsioon

Bakalaureusetöö eesmärk on luua kinnipidamisastustele prototüüp tarkvarast, mis võimaldab kinnipeetavatel isikutel luua digitaalselt pöördumisi või taotluseid kinnipidamisasutustele.

Töö käigus kavandatakse ning seejärel arendatakse veebirakendused, mis näitavad probleemi digitaalset lahendatavust. Bakalaureusetöö käigus kaardistatakse veebirakenduse nõuded ning põhjendatakse arenduses kasutatavate tehnoloogiate valikuid.

Ülesande edukal lahendamisel on planeeritud rakenduse arendamise jätkamine ning selle kasutusele võtmine kinnipidamisasutustes.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 30 leheküljel, 7 peatükki, 14 joonist, 7 tabelit.

Abstract

Prototype of Inmate Appeals Portal

The aim of this thesis is to create a prototype of software for detention facilities that allows detained persons to digitally create appeals or requests to detention facilities.

In the course of this thesis, applications necessary to fulfill the proposed solution are planned and then developed in order to demonstrate the digital solvability of the problem. The thesis maps the requirements of the web application and justifies the choices of technologies used in the development.

If the task is solved successfully, it is planned to continue the development of the application and eventually use it in detention facilities.

The thesis is in Estonian and contains 30 pages of text, 7 chapters, 14 figures, 7 tables.

Lühendite ja mõistete sõnastik

ACID	eelduste kogum, mis tagab kõigi andmebaasi tehingute usaldusväärse töötlemise, mille tulemuseks on korrektsus
AI	<i>artificial intelligence</i>
<i>boilerplate</i>	lähtekood, mis on kirjutatud rakenduse arendamise jooksul, et oleks võimalik luua funktsionaalsust
C#	tugevalt trükitud programmeerimiskeel
DTO	<i>data transfer object</i>
ERD	<i>entity relationship diagram</i> ehk olemi seoste diagramm
F#	funktsionaalne programmeerimiskeel
<i>front-end</i>	rakenduse kasutajaliides
gRPC	platvormideülene avatud lähtekoodiga suure jõudlusega kaugprotseduurikõnede (RPC) raamistik.
Javascript	ECMAScripti spetsifikatsioon baseeruv programmeerimis keel
JSON	<i>javaScript Object Notation</i>
ORM	<i>object relational mapping</i>
Python	üldotstarbeline programmeerimiskeel
RDBMS	<i>relational database management system</i>
REST	<i>representational state transfer</i>
RIK	registrite ja infosüsteemide keskus
SQL	<i>structured Query Language</i> ehk struktureeritud päringu keel
XML	<i>extensible Markup Language</i>
YAML	inimloetav andmete jadakeel.

Sisukord

1 Sissejuhatus	10
2 Metoodika.....	11
3 Probleem.....	12
3.1 Probleemi taust	12
3.2 Olemasolev lahendus	12
3.3 Planeeritav lahendus	13
3.4 Autori roll	15
4 Prototüübi äriloogika analüüs.....	16
4.1 Kasutajalood	16
4.2 Mittefunktsionaalsed nõuded.....	17
4.3 Pöördumise tegemine AS-IS	17
4.4 Pöördumise tegemine TO-BE.....	19
4.5 Rollid	19
4.6 Kinnipeetavate portaal.....	20
4.7 Vanglate portaal.....	20
4.8 API.....	21
4.9 Vastaja määramine ja vastuse andmine planeeritud infosüsteemis	22
4.10 ERD	22
5 Prototüübi tehniline analüüs	23
5.1 Veebivormi loomise võimalused kinnipeetavate portaalis	23
5.1.1 Content Management System.....	23
5.1.2 Flat-File Content Management System	23
5.1.3 Kinnipeetavate portaali kasutajaliidese tehnoloogiate võrdlus	24
5.2 Kinnipeetavate portaali ja vanglate portaali kasutajaliidese tehnoloogia valik....	24
5.2.1 TypeScript	25
5.3 API tehnoloogiate valik	25
5.3.1 API arhitektuuriline stiil	25
5.3.2 API tehnoloogia.....	26
5.4 Andmebaasi analüüs	26

5.4.1	Relatsioonilised andmebaasi haldussüsteemid	27
5.4.2	Mitte-relatsioonilised andmebaasi haldussüsteemid	28
5.4.3	Andmebaasisüsteemi valik	28
5.5	Pöördumise salvestamine infosüsteemis	29
5.6	Autentimine	30
5.6.1	JSON Web Token autentimine	31
6	Prototüübi arendusprotsess	32
6.1	API arendusprotsess	32
6.1.1	.NET versioon.....	32
6.1.2	Andmebaas	32
6.1.3	Arhitektuur	33
6.1.4	Esitlus	33
6.2	Kinnipeetavate portaali ja vanglate portaali arendusprotsess.....	35
6.2.1	Kinnipeetavate portaali vaated	35
6.2.2	Vanglate portaali vaated	37
6.3	Tulemused	40
7	Kokkuvõte	41
Lisa 1	– Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	45

Jooniste loetelu

Joonis 1. Pöördumiste esitamise senine struktuur [9].....	13
Joonis 2. Projekti arhitektuuriline visioon [6]	14
Joonis 3. Kasutajalood (Autori koostatud)	17
Joonis 4. Pöördumise tegemine AS-IS (Autori koostatud)	18
Joonis 5. Pöördumise tegemine TO-BE (Autori koostatud).....	19
Joonis 6. Rakenduse planeeritud ERD (Autori koostatud).....	22
Joonis 7. Swagger' vaade API otspunktidest	34
Joonis 8. Kinnipeetavate portaali esileht	35
Joonis 9. Kinnipeetavate portaali pöördumise loomise vaade.....	36
Joonis 10. Kinnipeetavate portaali esitatud pöördumiste vaade.....	37
Joonis 11. Vanglate portaali esileht	37
Joonis 12. Vanglate portaali dokumendihalduse vaade.....	38
Joonis 13. Vanglate portaali pöördumise registreerimine	39
Joonis 14. Vastuse andmine pöördumisele.....	40

Tabelite loetelu

Tabel 1. CMS-i ja Flat-File CMS-i võrdlus.....	24
Tabel 2. kaasaegse front-end raamistiku ja CMS-i võrdlus (autori koostatud).....	24
Tabel 3. Kasutajaliidese front-end raamistiku valik (autori koostatud)	24
Tabel 4. API tehnoloogiate valik (autori koostatud)	26
Tabel 5. Analüüs faili salvestamisest kõvakettale ning andmebaasi viide selle asukohast (autori koostatud).....	30
Tabel 6. Analüüs faili sisu salvestamisest andmebaasi (autori koostatud).....	30
Tabel 7. Analüüs faili salvestamisest, andmeväljadest koosneva andmebaasi kirjena (autori koostatud).....	30

1 Sissejuhatus

Vanglateenistuse ja vanglaametnikkonna põhiline eesmärk on kinnipeetavate inimeste karistuse täideviimine. Vanglaametniku töö ei ole tänapäeval okastraatide vahel jalutamine, püss õlal, vaid see tähendab mitmeplaani tööd inimestega [1].

Üheks vanglaametniku ülesandeks on kinnipeetavate isikute poolt esitatud dokumentide läbitöötamine, korrastamine ja vastuste vormistamine. Praeguste tavade järgi tuleb vangil vormistada ametlik pöördumine kui ei piisa ainult suulisest vastusest. Pöördumise esitamine on vajalik kui vangil on olmeprobleem, tervisemure või muu igapäeva eluolu puudutav küsimus [2]. Näiteks saab vang taotleda vangla direktorilt luba elektriseadme kasutamiseks, lühiajaliseks väljasõiduks või pikaajaliseks kokkusaamiseks. Taotlus kas lahendatakse ja isik saab positiivse või negatiivse vastuse, või tagastatakse läbi vaatamata seaduses toodud aluste esinemisel [3]. Lisaks võib vangla poole pöörduda peale kinni peetavate ka vanglavälised isikud.

Vanglaametnik on kohustatud vajadusel välja selgitama, mis oli isiku tegelik tahe ning abistama isikut oma tahte väljendamisel. Lisaks võib pöördumisele olla vajalik anda vastus mitme osapoole poolt, kes võivad asuda erinevates struktuuriüksustes [4]. Paber kandjal erinevatele pöördumistele vastuseandmine võtab täna vanglaametnikult märkimisväärse aja. Puhtalt logistiliselt on praegune lahendus vanglaametnikele ebavajalikuks koormaks.

Käesoleva bakalaureusetöö käigus luuakse vanglaametnikkonnale prototüüp lahendusprobleemi digitaalselt lahendamiseks. Esmalt viiakse läbi ärianalüüs kuhu kaasatakse vanglatöötajad ning mille käigus selgitatakse välja ärioloogilised vajadused. Seejärel teostatakse süsteemianalüüs, mis seisneb kavandatava rakenduse arhitektuuriliste ja tehnoloogiliste vajaduste kaardistamises. Viimaks realiseerib autor rakenduse prototüüplahenduse, millega tõestatakse probleemi lahendatavust digitaalselt.

2 Metoodika

Diplomitöö käigus esmalt tutvustatakse olemasolevat probleemi. Selgitatakse lahenduse vajalikkust. Kirjeldatakse probleemi tausta ning demonstreeritakse olemasolevat lahendust.

Järgnevalt teostatakse põhjalik ärianalüüs, kus antakse täpne ülevaade ärioloogilistest nõuetest. Kaardistatakse põhiprotsess. Vastatakse kuidas toimub põhiprotsess täna ning toetudes autori poolt koostatud kasutajalugudele, määratletakse kuidas põhiprotsess saab olema rakendust kasutades.

Pärast mida sooritatakse mitmekülgne süsteemianalüüs. Süsteemianalüüsi käigus tuginetakse eelnevalt tehtud ärianalüüsile ning pakutakse arhitektuuriline lahendus. Lisaks võrreldakse erinevaid tehnoloogiaid, mis võimaldavad prototüüplahendust teostada ning valitakse sobivaimad.

Praktiline osa hõlmab rakenduse komponentide loomist ja seejärel kavandatava lahenduse detailide välja arendamist. Prototüübi arendusprotsess tugineb eelnevalt määratud teoreetilise osa lähtepunktidele.

Viimaks tehakse loodud infosüsteemist kokkuvõtte. Analüüsitakse saavutatud tulemust ning vaadatakse üle tulemuse vastavus nõuetele. Antakse hankija poolne tagasiside. Juhul kui saavutatud tulem on ootuspärane ning tellija jaoks aktsepteeritav, siis kirjeldatakse planeeritav arendusplaan rakenduse täielikuks realiseerimiseks

3 Probleem

Täna ei ole kinnipeetaval võimalik, esitada kinnipidamisasutuses elektrooniliselt taotluseid või pöördumisi, ega tutvuda dokumentidele koostatud vastustega. Vangi jaoks on terve protsess paber kandjal. Elektrooniliselt taotluste ja pöördumiste haldamine võimaldaks vanglaametnikul muuta pöördumiste andmevälju, mis on kinnipeetava poolt väärtalt täidetud. Luua ja muuta erinevaid kinnipeetavatele suunatud vorme. Digitaalne lahendus kiirendaks vastuse andmist näiteks juhtudel, kus pöördumisele vastuse andmiseks on vajalik mitmel osapoolel dokumendiga tutvuda. Lisaks võib kinnipeetav teha pöördumisi ka mõnele teisele haldusorganile.

3.1 Probleemi taust

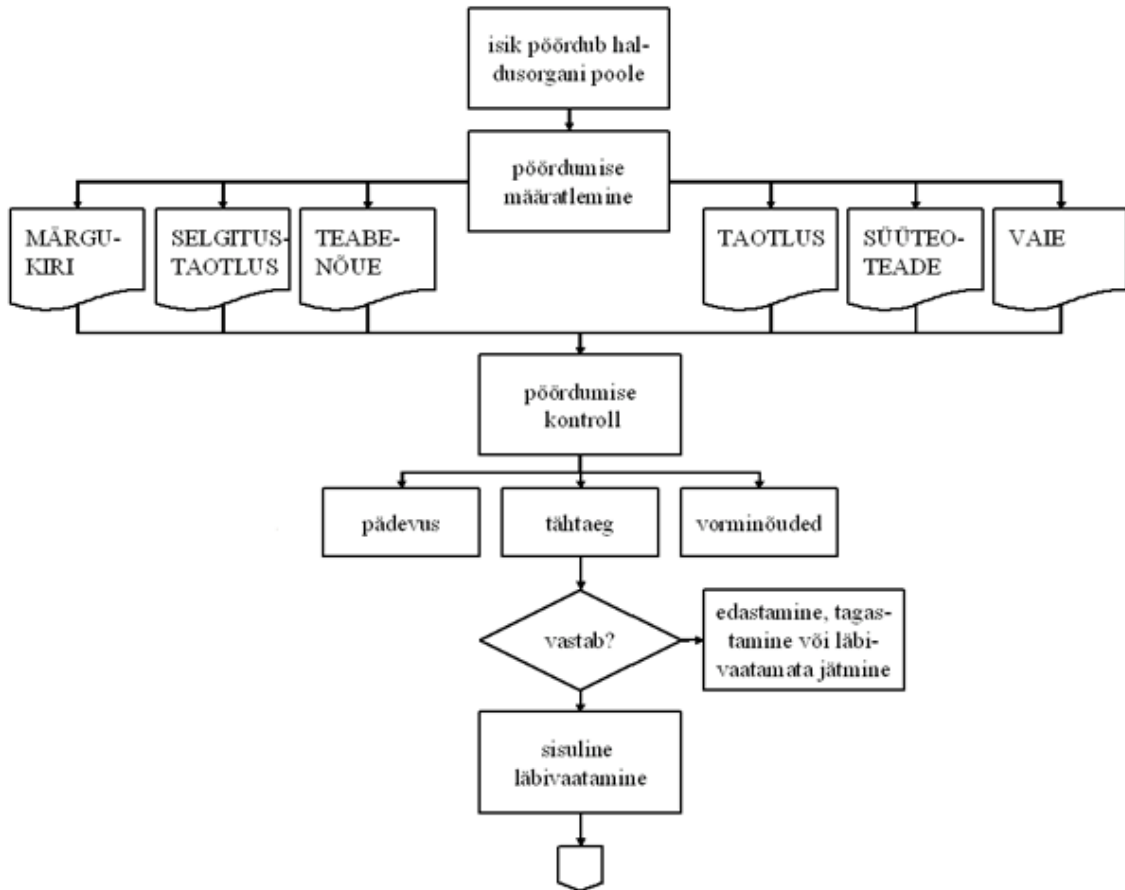
Eesti vanglates on ligikaudu 2250 kinnipeetavat [5]. Registreeritud pöördumiste suurusjärk on kümnetes tuhandetes. Vanglaametnike töökoormuse vähendamiseks ning tõhusama töökorralduse loomiseks on RIK-i poolt arendamisel veebikeskkond, mis on mõeldud vangil karistuse kandmisega seotud e-teenuste kasutamiseks. Erinevate teenuste koondamiseks luuakse kinnipeetavatele isikutele töölaud, mis sisaldab viiteid erinevatele e-teenustele, mida kinnipeetavale on lubatud tarbida. Üheks nendest kavandatavatest rakendustest on „Kinnipeetavate isikute portaal“, mis omakorda koosneb kahest eraldi portaalist „Kinnipeetava portaal“ ja „Vangla portaal“ [6]

3.2 Olemasolev lahendus

Üldjoontes on hetkene praktika järgnev:

- 1) Kinnipeetav isik teeb paber kandjal pöördumise ning annab selle kas endale määratud inspektor-kontaktisikule või ettejuhtuvale vangla valvurile [7]. Kusjuures ei ole ebatavaline, et avaldus ei ole kirjutatud puhtale valgele paberile vaid on kasutatud hoopis näiteks õpikulehte.
- 2) Kui taotlus on arusaadav ja ei nõua täiendamist, siis skaneeritakse dokument ning sisestatakse käsitsi dokumendihaldussüsteemi Delta [8].

- 3) Seejärel antakse Deltas dokumendile vastus ning toimetatakse välja prinditud dokument vangile.
- 4) Vang nõustub või ei nõustu vastusega. Vangi nõusolek fikseeritakse vanglates erinevalt.



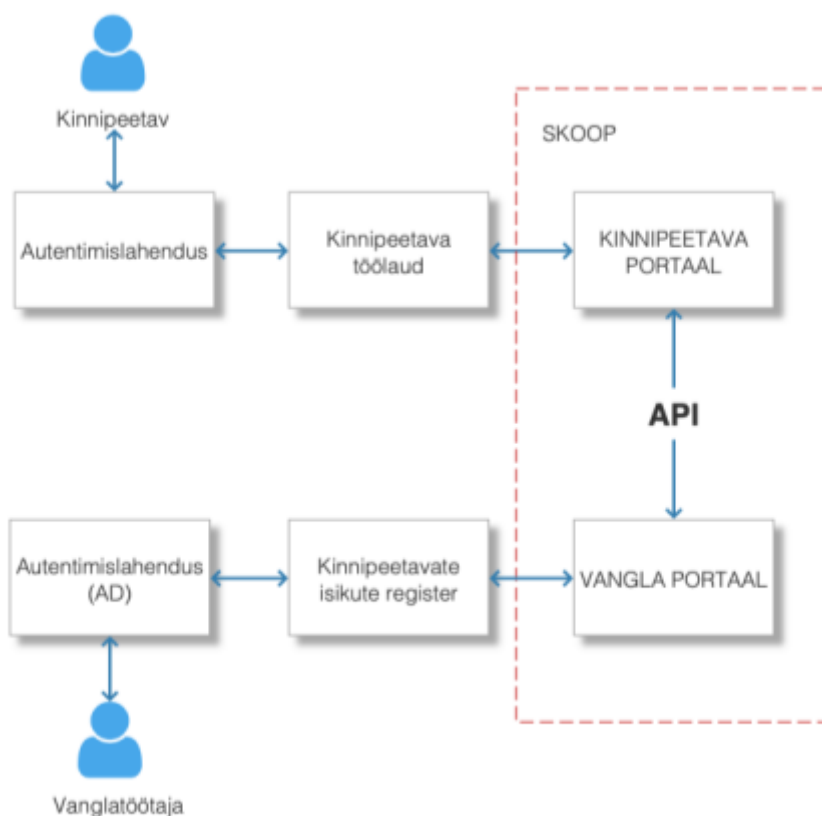
Joonis 1. Pöördumiste esitamise senine struktuur [9]

3.3 Planeeritav lahendus

Diplomitöös arendatav prototüüplahendus koosneb kolmest komponendist. Esimene komponent on „Kinnipeetavate isikute portaal“. Kinnipeetava portaal on rakendusele kinnipeetavale rakenduse esiliides, mille vahendusel on kinnipeetaval võimalik esitada taotluseid või pöördumisi, dokumentide staatustega tutvuda ning dokumentidele koostatud vastustega tutvuda. Teine komponent on „Vanglate portaal“. Vanglate portaal on rakendusele haldusorgani poolne esiliides, mis on mõeldud kinnipeetavate poolt esitatud pöördumiste redigeerimiseks, registreerimiseks, vastaja määramiseks ning

vastuse andmiseks. Kolmas komponent on API, mis teostab äriloogikat ning võimaldab suhtluse planeeritavatele portaalidele.

Prototüüp lahendusena ei ole vajalik dokumentatsiooni, kujundus jm viimistletus, kuid lahendus peab töötama määral, mis võimaldab tellijal veenduda lahendi ärilises sobivuses, tehnilises teostatavuses ja jätkusuutlikkuses [6]. Joonisel 2 on tellija sisend esimene nägemus projekti arhitektuurist



Joonis 2. Projekti arhitektuuriline visioon [6]

3.4 Autori roll

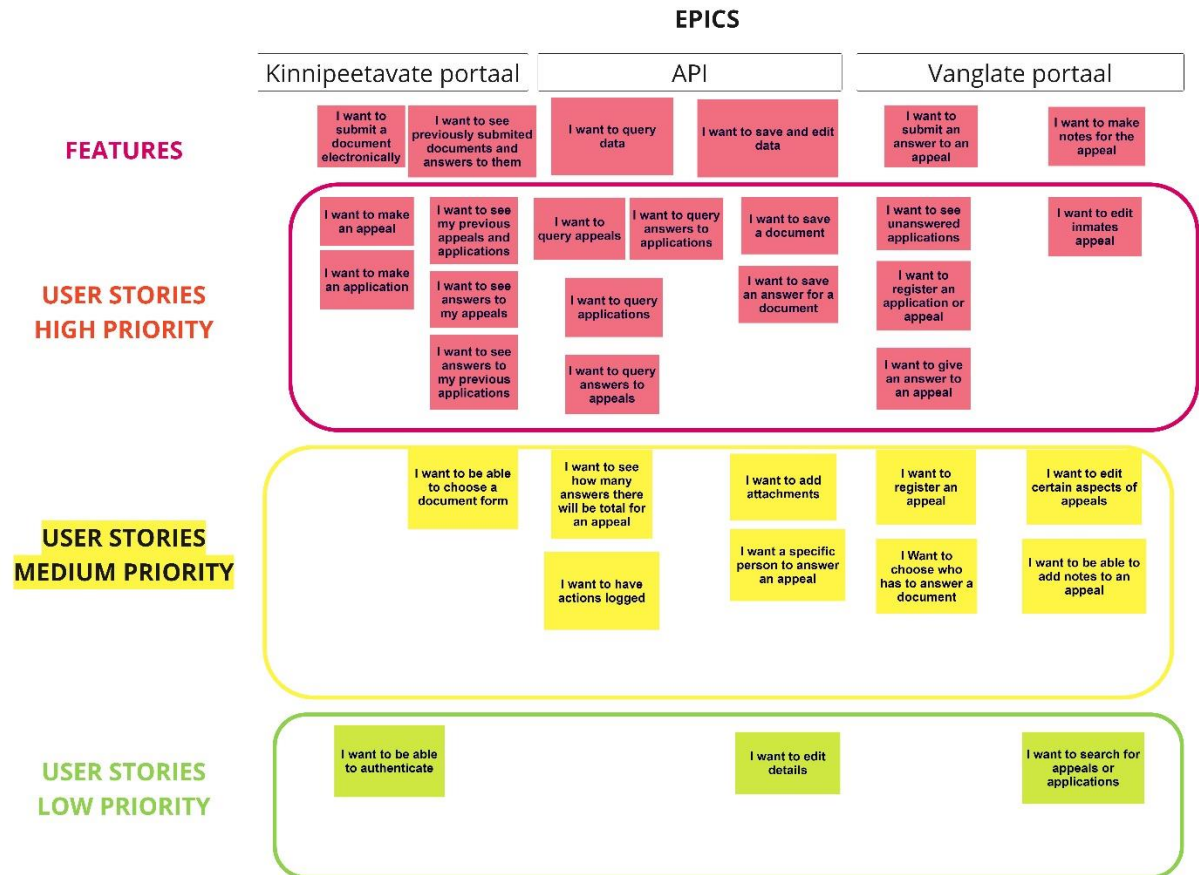
Bakalaureusetöö autor töötab Registrate ja infosüsteemide keskuse nooremarendaja ametikohal ning tegeleb igapäevaselt Justiitsministeeriumi vastutusalas tegutseva Avaliku E-toimiku arendamisega. Autor on osalenud vangidele mõeldud ning modifitseeritud funktsionaalsuse integreerimisel avaliku e-toimikusse.

4 Prototüübi ärioloogika analüüs

Peatükk keskendub peamiselt rakenduse ärivajadustele. Peatükis koostatakse kasutajalood, kirjeldatakse põhiprotsessi ning kaardistatakse nõuetest tingitud vajalik funktsionaalsus. Põhjalik ärianalüüs on vajalik potentsiaalsete probleemide paljastamiseks [10].

4.1 Kasutajalood

Kasutajalood on arendusprotsessis kasulik meede, sest need aitavad tabada kasutaja vajadusi ja kindlaks määrata nõudeid. Andes üksikasjaliku kirjelduse selle kohta, kuidas kasutaja kavatseb rakendust kasutada. Tulemusena luuakse kasutajakeskne toode [11]. Lisaks on diplomitöö raames koostatud kasutajalood vajalikud ka, et võimalikke probleeme enne nende tekkimist ette näha, tagades nii eduka arendustsükli. Autori poolt koostatud kasutajalood on kirjeldatud järgneval joonisel.



Joonis 3. Kasutajalood (Autori koostatud)

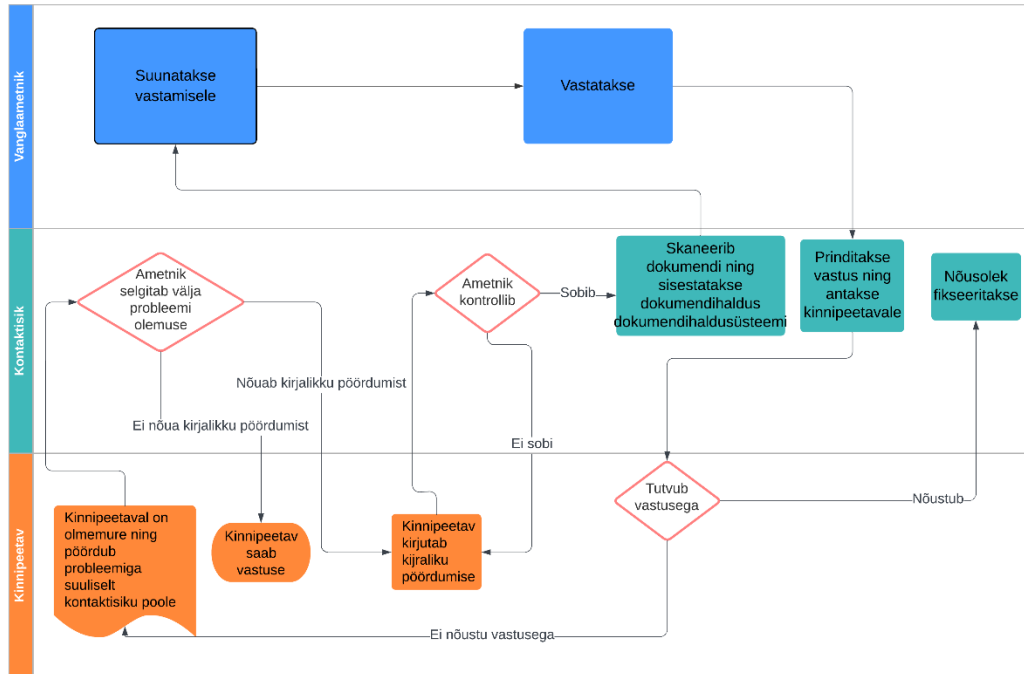
4.2 Mittefunktsionaalsed nõuded

Kasutatava platvormi valimisel on Hankija jaoks oluline, et [6]:

- 1) Vormide administreerimine peab olema niivõrd lihtne, et selle teostamiseks ei ole vaja eriteadmisi
- 2) Platvorm on edasiarendatav hankija poolt, st lähtekood on avalik
- 3) Platvormi võimaldaks luua tänapäevase väljanägemise
- 4) Pakkuda head kasutajakogemust

4.3 Pöördumise tegemine AS-IS

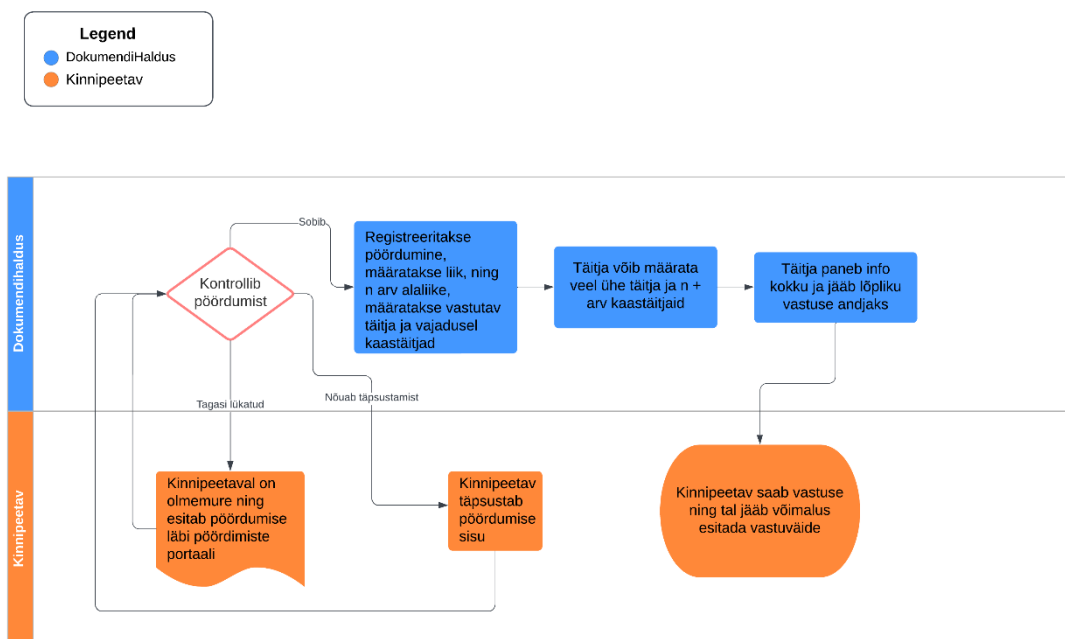
Järgneval joonisel on kirjeldatud pöördumise esitamise põhiprotsess ilma infosüsteemi olemasoluta.



Joonis 4. Pöördumise tegemine AS-IS (Autori koostatud)

4.4 Pöördumise tegemine TO-BE

Järgneval joonisel on kirjeldatud pöördumise esitamise põhiprotsess planeeritavat infosüsteemi kasutades.



Joonis 5. Pöördumise tegemine TO-BE (Autori koostatud)

4.5 Rollid

Pöördumise käitlemisel on kavandatavas infosüsteemis selgelt piiritletud vastutusosalad. Vastutusosaladest saame umbkaudselt paika panna tulevased rollid. Kinnipeetavate portaali funktsionaalsus jääb kasutada ainult rollile “vang” ning ülejäänud manageerimine, mis on teostatav vanglate portaalis, kuulub ainult rollile “dokumendihaldus”. Lisaks võib olla tarvidus kasutusele võtta IT-alaseid teadmisi omavale, nii-öelda haldurile, roll “admin”.

4.6 Kinnipeetavate portaal

Kinnipeetava portaali vahendusel on võimalik kinnipeetaval teha erinevaid taotluseid või pöördumisi kinnipidamisasutustele ning nende vastustega tutvuda. Näiteks kokkusaamise taotlus, teabenõue väljasõidu taotlus, vara kambrisse/lattu paigutamise taotlus jne [6].

Kinnipeetavate portaalile planeeritud funktsionaalsus:

- 1) Uue pöördumise koostamine
- 2) Pöördumise esitamine
- 3) Esitatud pöördumiste kuvamine
- 4) Pöördumise vastuse kuvamine

4.7 Vanglate portaal

Kinnipeetavate poolt sisestatud pöördumistele vastamiseks tuleb luua eraldi esiliides vanglatöötajatele ehk vanglate portaal [6].

Vanglate portaali planeeritud funktsionaalsus:

- 1) Esitatud pöördumiste kuvamine
- 2) Esitatud pöördumiste andmeväljade muutmine
- 3) Pöördumise registreerimine
- 4) Pöördumisele märkmete lisamine
- 5) Pöördumisele manuse lisamine
- 6) Vastajale vastamiseks suunatud pöördumiste kuvamine
- 7) Pöördumisele sisuline vastamine
- 8) Pöördumiste otsing

4.8 API

API-sid kasutatakse infosüsteemides erinevate komponentide vaheliseks suhtluseks. API loob turvalise ja usaldusväärse kihi andmete juurdepääsul. API-d lihtsustavad erinevate süsteemide integreerimist ning tulevikus täiendava funktsionaalsuse lisamist. Lisaks võimaldab selline arhitektuuriline lähenemine andmete pärimiseks kasutada mõnda teist masinliidest [12].

API planeeritud funktsionaalsus:

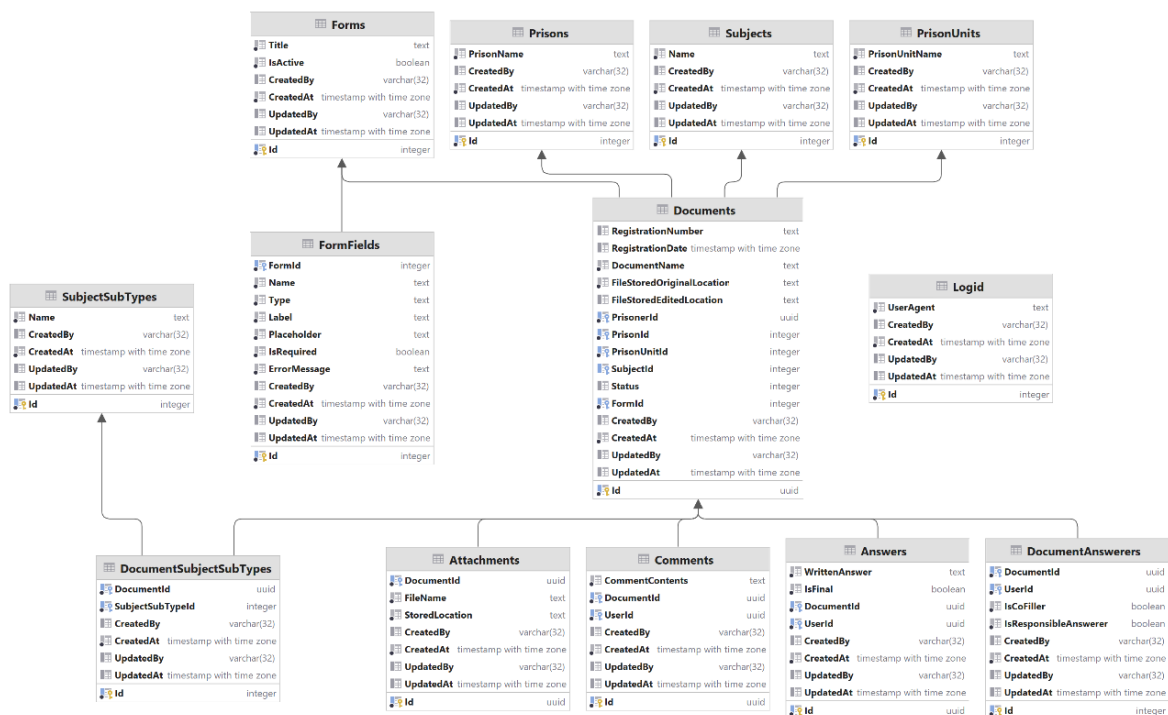
- 1) Pöördumiste salvestamine
- 2) Pöördumiste pärimine
- 3) Pöördumise registreerimine
- 4) Pöördumisele märkmete lisamine
- 5) Pöördumine vastamisele suunamine
- 6) Pöördumisele vastaja määramine
- 7) Pöördumisele vastuse lisamine
- 8) Pöördumisele või vastusele manuse lisamine
- 9) Pöördumisele vastuste koguarvu lisamine
- 10) Kinnipeetava vastus(t)ega tutvumise kuupäeva ja kellaaja logimine
- 11) Tegevuse logimine
- 12) Veateate logimine

4.9 Vastaja määramine ja vastuse andmine planeeritud infosüsteemis

Dokumendihaldus määrab vastutava täitja ja vajadusel kaastäitjad. Täitja võib omakorda määrata veel ühe täitja ja lõputult kaastäitjad. Näiteks võib dokumendihaldus määrata täitjaks osakonnajuhataja, aga temal on delegerimiseks õigus täitja roll edasi määrata teisele isikule. Sellisel juhul liiguvad kaastäitjad automaatselt kaasa. Täitja paneb info kokku ja jääb lõpliku vastuse andjaks. Kõikidel täitjatel jääb allkirjastamise õigus ning võimalus suunata kolmandatele osapooltele.

4.10 ERD

Järgneval joonisel on autori poolt koostatud infosüsteemi andmebaasi relatsiooniline mudel.



Joonis 6. Rakenduse planeeritud ERD (Autori koostatud)

5 Prototüübi tehniline analüüs

Järgnev peatükk keskendub peamiselt projekti tehnilistele aspektidele. See hõlmab erinevate tehnoloogiate ja tehnikate sobivuse analüüsimist, mida süsteemi ehitamiseks on vajalik kasutada. Lisaks analüüsitakse vajalikke protseduure, mis tagavad süsteemi nõuetekohase töötamise.

5.1 Veebivormi loomise võimalused kinnipeetavate portaalis

Tehnoloogia valiku langetamisel tuleb arvesse võtta, et vorme oleks võimalik luua eriteadmisteta. Diplomitöös kaalutakse vormide loomiseks järgnevaid võimalusi:

- 1) Kasutada vabavaralist *Content Management System*'it
- 2) Kasutada vabavaralist *Flat-File Content Management-System*'it
- 3) Kasutada modernset esiliidese raamistikku ning integreerida vormid abstraktselt, et neid oleks võimalik lisada ja kasutusele võtta rakenduse siseselt.

5.1.1 Content Management System

CMS ehk sisuhaldussüsteem tarkvara, kasutatakse tüüpiliselt digitaalse sisu loomiseks, haldamiseks ja muutmiseks. Tugevaks eeliseks CMS-idel on pakutavad tööriistad kasutajatele sisu loomiseks ja haldamiseks. Näiteks on CMS-ide abil võimalik luua veebilehekülgi ilma suuremaid IT-alaseid teadmisi omamata. Lisaks pakuvad CMS-id sageli valmislahendustena funktsioone nagu näiteks kasutaja autentimine ja otsingumootori optimeerimine [13].

5.1.2 Flat-File Content Management System

Flat-file sisuhaldussüsteem (CMS) on CMS-i tüüp, mis salvestab sisu ja andmeid näiteks teksti failidesse, mitte andmebaasi. *Flat-file* CMS-e on lihtsam seadistada ja hooldada, sest pole näiteks tarvidust andmebaase konfigurereida ega andmebaasi tabelid hallata. Lisaks on *Flat-file* CMS-id sageli kiiremad kui traditsioonilised andmebaasid, sest andmed salvestatakse ühes failis ja need ei nõua keerulisi päringuid [14].

5.1.3 Kinnipeetavate portaali kasutajaliidese tehnoloogiate võrdlus

Kinnipeetavate portaali kasutajaliidese tehnoloogiate paika panemiseks analüüsitakse esmalt CMS-i ja Flat-File CMS-i ning seejärel analüüsitakse CMS-i kasutamist modernse Javascript-i raamistikuga.

Tabel 1. kaasaegse front-end raamistiku ja CMS-i võrdlus (autori koostatud)

CMS	Flat-File CMS
Eelised	-
Rohkem kasutajaid = rohkem abimaterjale	Vormid on inimesele loetavamas tekstivormingus näiteks: Markdown
Rohkem pluginaid	Hinnanguliselt on administreerimine lihtsam
Puudused	-
Vormide loomine nõuab HTML-i tundmist	Keerulise funktsionaalsuse loomine nõuab suuremat arendusressurssi

Tabel 2. CMS-i ja Flat-File CMS-i võrdlus

Moderne Javascript-i raamistik	CMS
Eelised	-
Võimaldab rohkem kohandamist	Madalamad hoolduskulud
Attraktiivsem arendajale	Administreerimine on oluliselt lihtsam
Puudused	-
Nõuab rohkem esialgset arendusressurssi	Keerulisem disainida kasutajasõbralikku ja omanäolist kasutajaliidest

Autori hinnangul on veebirakenduse arendamine, kasutades mõnda modernset Javascript'il baseeruvat raamistiku, vähem arendusressurssi nõudlik kui näiteks kasutada CMS-i. Lisaks toob CMS-i kohastamine nõutud funktsionaalsusega, suure tõenäosusega kaasa moodulite kirjutamise arendajale võõras programmeerimiskeeles.

5.2 Kinnipeetavate portaali ja vanglate portaali kasutajaliidese tehnoloogia valik

Kaalutud tehnoloogiate nimistusse kuuluvad React, Angular ning Vue, sest kõik kaalutud tehnoloogiad on RIK-is asutuse siseselt kasutuses.

Tabel 1. Kasutajaliidese front-end raamistiku valik (autori koostatud)

Raamistik	Kogemus	Õppimise keerukus	Nõuetele vastavus
React	Hea	Keskmine	Hea
Angular	Puudub	Keskmine	Hea
Vue	Madal	Keskmine	Hea

Analüüsi järgselt langetas autor otsuse kasutada projekti jaoks Javascript'i raamistikku React. Autor on raamistikuga varem töötanud ning selle süntaksi ja arendusprotsessiga tuttav. Raamistik sobib hästi projekti vajadustega ning tagab süsteemile vajaliku paindlikkuse. Täiendavalt võimaldab raamistik süsteemi kiiresti prototüüpida [15].

5.2.1 TypeScript

TypeScript on avatud lähtekoodiga, tugevalt trükitud JavaScripti superkomplekt. See on programmeerimiskeel, mis lisab JavaScriptile valikulisi staatilisi tüüpe, mis võimaldab arendajatel tõhusamalt koodi kirjutada [16]. Autori hinnangul on hea kasutada React'i koos TypeScript'iga

5.3 API tehnoloogiate valik

Veebi API on programmiline liides, mis võimaldab kahel rakendusel üksteisega Interneti kaudu suhelda, tavaliselt kasutades HTTP-protokolli. Veebi API-d pakuvad juurdepääsu veebiteenustele ja tagastavad andmeid ühtsetes vormingutes.

5.3.1 API arhitektuuriline stiil

API arhitektuuristiilideks kaalutakse projektis REST (Representational State Transfer) ja RPC (Remote Procedure Calls).

REST on API arhitektuuristiil hajutatud süsteemide ehitamiseks. REST API-d kasutavad HTTP-tegusõnu nt GET, POST, PUT, DELETE, et teostada API-le määratud funktsionaalsust. Andmetevahetamiseks kasutatakse JSON või XML [16].

gRPC on suure jõudlusega avatud lähtekoodiga kaugprotseduurikõne (RPC) raamistik, mis kasutab HTTP/2 protokolli. Andmete vahetamisel kasutatakse protokollipuhvrit, mis on on Google'i loodud binaarne serialiseerimisvorming, mida kasutatakse struktureeritud andmete vahetamiseks erinevate teenuste vahel. Protokoll on keele- ja platvormineutraalne [18].

Lihtsuse tõttu on kavandatavale infosüsteemile inspiratsiooniks RESTful API arhitektuuriline stiil, sest arendatav prototüüp on omanäoline ja toores. Stiil võimaldab kohandada koostatud nõuetele ja vajadustele, kasutades stiili paindlikkust. Samuti võimaldab see vajaduse korral süsteemi kiiresti muuta või värskendada, ilma et peaks arhitektuuriliselt keerulisemaid otsuseid vastu võtma.

5.3.2 API tehnoloogia

API tehnoloogiate valikul kaalutakse kolme raamistikku: .NET, Django ja node.js.

.NET raamistik on võimas ja usaldusväärne platvorm. See toetab mitut keelt, sealhulgas C# ja F#. Samuti pakub see juurdepääsu pakettide teegile, mida saab kasutada API funktsionaalsuse laiendamiseks. .NET Framework pakub mitmeid valmislahendusi turbemehhanismide juurutamiseks [18].

Node.js-i on lihtne seadistada ja hooldada ning see ühildub paljude populaarsete veebikeeltega. Samuti pakub see juurdepääsu suurele pakettide teegile, mis lihtsustab API funktsionaalsuse laiendamise. Node.js on skaleeritav, mistõttu sobib see rakendustele, mis peavad käsitlema suuri andmemahutusi [19].

Django on tasuta ning avatud lähtekoodiga Python'il põhinev veebiraamistik, mis järgib Model-View-Template (MVT) arhitektuurimustrit. Raamistiku eeliseks on kiire kasutusele võtmine ning vähene vajalik *boilerplate*, mis aitab keskenduda rakenduse kirjutamisele [21].

Tabel 2. API tehnoloogiate valik (autori koostatud)

Raamistik	Programmeerimiskeel	Kogemus	Õppimise keerukus	Nõuetele vastavus
.NET Framework	C#	Väga hea	Keskmine	Hea
Node.JS	Javascript	Madal	Madal	Hea
Django	Python	Keskmine	Madal	Hea

Nõuetele vastavus on kõigil raamistikel sama. Arendajal on põhjalikum kogemus .NET Framework raamistikul. Lisaks on Registrate ja Infosüsteemide Keskusel .NET raamistikul töötavatele rakendustele X-teel suhtlemise toetamiseks loodud standardmoodul.

5.4 Andmebaasi analüüs

Andmebaasihaldussüsteemi (DBMS) valimine on iga rakenduse jaoks oluline otsus. Kuna erinevatel rakendustel on unikaalsed funktsioonid, siis tehnoloogiate valimisel ei ole ainuõiget otsust toote arendamiseks [22]

5.4.1 Relatsioonilised andmebaasi haldussüsteemid

Struktureeritud päringukeele ehk SQL on andmebaasiga suhtlemise keel, mis on loodud andmete haldamiseks relatsiooniliste andmebaaside haldussüsteemides. SQL hõlmab päringuid, andmete sisestamist, kustutamist, värskendamist, skeemi loomist ja muutmist [23]

Relatsiooniliste andmebaasisüsteemide eelised [24]:

- ✓ Kasutuslihtsus
- ✓ Stabiilsus ja usaldusväarsus
- ✓ Varundamise ja avariitaaste lihtsus

5.4.2 Mitte-relatsioonilised andmebaasi haldussüsteemid

NoSQL-i andmebaasid on muutunud populaarseks, kuna need võimaldavad salvestada mitmetest allikatest pärinevaid andmeid, mis kasutavad erinevaid vorminguid. Andmete salvestamisel või mälust väljavõtmisel on vaja teha vähem teisendusi. NoSQL-i andmebaasi võimaldab salvestada palju erinevaid andmetüüpe, olgu need struktureerimata, struktureeritud või poolstruktureeritud [23, 25].

Mitte-relatsiooniliste andmebaasisüsteemide eelised [26]:

- ✓ Ideaalne struktureerimata andmete jaoks
- ✓ AI-algoritmidele paremini analüüsitav
- ✓ „Suurandmete“ analüütika käsitlemiseks
- ✓ Salvestatavate andmetüüpide osas pole piiranguid

5.4.3 Andmebaasisüsteemi valik

Andmebaasisüsteemid, mida kaalutakse rakenduses kasutamiseks on Oracle ja PostgreSQL. Mõlemad tehnoloogiad on hetkel RIK-is kasutusel.

Oracle andmebaas on objektrelatsioonilise andmebaasihaldussüsteem. Oracle RDBMS salvestab andmeid tabeliruumide ja segmentide kujul, näiteks nagu andmesegmendid, indeksisegmendid jne. Segmendid omakorda sisaldavad ühte või mitut *extent*'i ehk ulatust. Ulatused koosnevad külgnevate andmeplokkide rühmadest. Andmeplokkid loovad andmete salvestamise põhiühikud [23].

Oracle andmebaasi iseloomustab [23]:

- ✓ Laiihaardelisuus
- ✓ Kõrgtehnoloogilisuus
- ✓ Kõrge pingetaluvusega
- ✓ Kulukas

PostgreSQL on avatud lähtekoodiga objekt-relatsiooniline andmebaasisüsteem, mis kasutab ja laiendab SQL-keelt koos paljude funktsioonidega. PostgreSQL on pälvinud tugeva maine oma tõestatud arhitektuuri, töökindluse, andmete terviklikkuse, tugeva funktsioonide komplekti, laiendatavuse ja tarkvara taga oleva avatud lähtekoodiga. PostgreSQL töötab kõigis suuremates operatsioonisüsteemides, on ACID-ühilduv alates 2001 [27]. PostgreSQL andmebaasi iseloomustab:

- ✓ Toetab nii relatsioonilisi kui ka mitte relatsioonilisi andmevorminguid [28]
- ✓ Tasuta
- ✓ Vabavaraline

Projekti arendamisel kasutatakse PostgreSQL-i, sest tarkvara on tasuta, vabavaraline. Tulevikukindluse mõttes on ka relatsiooniliste kui ka mitte relatsiooniliste andmevormingute toetamine atraktiivne viimasel ajal AI-algoritmide plahvatusliku arengu tõttu.

5.5 Pöördumise salvestamine infosüsteemis

Autori hinnangul on vormi sisu salvestamise jaoks kolm lahendust, mis sobivad hästi loodavasse infosüsteemi.

- ✓ Salvestada fail kõvakettale ning andmebaasi viide selle asukohast
- ✓ Salvestada faili sisu andmebaasi
- ✓ Salvestada andmeväljadest koosnev olem andmebaasi

Fail kõvakettal ning andmebaasis on viide selle asukohast analüüs:

Tabel 3. Analüüs faili salvestamisest kõvakettale ning andmebaasi viide selle asukohast (autori koostatud)

Eelised	Puudused
Faili kõvakettale salvestamine soodustab süsteemi administreerimist ilma eriteadmisteta.	Esineksid potentsiaalsed turvaprobleemid. Failide lugemine teiste süsteemide poolt eeldab, et fail on salvestatud süsteemi mõttes avalikule kettale
Andmebaasi viitesele on võimalik lisada lisaandmeid faili kohta.	Väiksem skaleeritavus
	Hooldatavus on küll põhimõtteliselt lihtsam, kuid kui midagi valesti peaks minema, siis sel juhul süsteem ei pruugi korrektselt enam toimida

Faili sisu salvestatakse andmebaasi analüüs:

Tabel 4. Analüüs faili sisu salvestamisest andmebaasi (autori koostatud)

Eelised	Puudused
Andmete lugemine on lihtsam	Andmebaasi haldamise keerukus
	Päringute kirjutamine on keerulisem ning on suurem tõenäosus, et süsteem peab filtreerimiseks tegema palju nõ "tühja tööd"
	Andmebaasi koormus faili lugemisel
	Failide lugemine on tehniliselt keerukam

Andmeväljadest koosnev andmebaasi kirje analüüs:

Tabel 5. Analüüs faili salvestamisest, andmeväljadest koosneva andmebaasi kirjena (autori koostatud)

Eelised	Puudused
Pöördumiste andmeväljade muutmine on oluliselt lihtsam	Universaalne lahendus on tõenäoliselt võimatu, see tähendab, et uue vormi lisamisel on tõenäoliselt vaja süsteemi kohandada (lisada uusi DTO-si jne).
Süsteem on kiirem	API nõuab rohkem arendusressurssi (ERD schema, spetsiifilised DTO-d, rohkem otspunkte)

5.6 Autentimine

Projekti skoobi sisse ei ole planeeritud autentimislahenduse välja arendamine küll aga, nõuab projekt äri loogika raames autentimist. Selle tõttu on esialgne autentimislahend integreeritud projekti algusest peale.

5.6.1 JSON Web Token autentimine

JSON Web Token (JWT) autentimine on kasutajate autentimise viis. JWT on avatud tööstusstandardi meetod turvaliseks andmevahetuseks kahe osapoole vahel. See on turvaline ja olekuta autentimismehhanism, milles ei salvestata kasutaja sessiooni serveri mälli vaid valideeritakse rakenduse poolt välja antud nii-öelda võtit. Tokenid allkirjastatakse kas salajase või avaliku/privaatse võtmepaari abil. Kasutaja edukal sisse logimisel, luuakse ja saadetakse kliendile JSON Web Token, mis sisaldab sellist teavet nagu kasutaja identiteet ja muud nõuded. Seejärel saadab klient loa serverile tagasi iga järgneva päringu jaoks, võimaldades serveril kasutaja identiteeti kontrollida ja päringuid autoriseerida [29]. RIK-is on mitmetesse rakendustesse juba integreeritud JWT kasutamine.

Arendusprotsessis kasutatakse .NET teeki Microsoft.AspNetCore.Authentication.JwtBearer, et funktsionaalsus integreerida.

6 Prototüübi arendusprotsess

Käesolev peatükk käsitleb prototüübi arendus metoodikaid ning arendusprotsessis olnud märkimisväärseid takistusi. Peatüki eesmärk on demonstreerida põhjalikult rakenduse loomist.

6.1 API arendusprotsess

API arendusprotsessis kirjeldatakse API-s kasutatava raamistikuga rakenduse loomist.

6.1.1 .NET versioon

Rakenduse arendamisel on kasutatud uusimat .NET versiooni. .NET versioon 7 pakub palju uusi funktsioone ja täiustusi võrreldes eelmiste versioonidega. Microsoft'i poolt välja toodud suurimad edusammud on tehtud valdkondades: täiustatud jõudlus, täiustatud vaadeldavus, täiustatud regulaaravaldised ning uuendatud .NET-i teegid [30].

6.1.2 Andmebaas

Andmebaasi loomiseks on projektis kasutatud Docker Compose. Docker Compose on tööriist, mis on välja töötatud mitmik-konteiner rakenduste loomiseks ja jagamiseks [31]. Arendamisel on loodud vajaliku sisuga docker-compose YAML-fail, et oleks võimalik ühe käsuga andmebaas läbi Docker-i luua ja käivatada.

Projektis kasutatakse Entity Framework Core andmebaasi kasutamiseks rakenduses. Entity Framework Core on tarkvara, mis on loodud tõlkima andmebaasides kasutatavate ja objektorienteeritud programmeerimises kasutatavate andmete esitusviise [32].

Andmebaasi tabelite loomiseks on kasutatud EF Core'i migreerimisfunktsiooni. EF Core'i migreerimisfunktsioon võimaldab andmebaasi skeemi järk-järgult värskendada, et hoida seda sünkroonis rakenduse andmemudeliga, säilitades samal ajal andmebaasis olevad andmed [33]. Migratsioonid toimivad järgmiselt. Andmemudeli muudatuse sisseviimisel kasutab arendaja EF Core'i tööriistu, et lisada vastav migratsioon, mis kirjeldab andmebaasi skeemi. EF Core võrdleb praegust mudelit vana mudeli hetktõmmisega, et teha kindlaks erinevused, ja genereerib migratsiooni lähtefailid. EF


Core salvestab kõik rakendatud migratsioonid spetsiaalsesse ajalootabelisse, mis võimaldab sellel teada, millised migratsioonid on rakendatud ja millised mitte [34].



6.1.3 Arhitektuur

Rakendus on planeeritud olema kihilise arhitektuuristiiliga. Kihiline arhitektuuristiil on üks levinumaid arhitektuuristiile. *Layered Architecture* idee seisneb selles, et sarnaste funktsioonidega moodulid või komponendid on korraldatud horisontaalsetesse kihtidesse [35]. Kihid on rakenduses nimetatud järgnevalt: *Data access layer*, *business-logic layer* ja *Presentation layer*. Kihtide vaheline suhtlus käib läbi DTO-de ning et protsess ei nõuaks käsitsi objektide kaardistamist kasutatakse rakenduses andmebaasi domeeniobjektide käitlemisel teeki AutoMapper. AutoMapper on loodud, et objektide teisendamist lihtsutada ning automatiseerida [36]. Lisaks on rakenduse siseselt praktiseeritud varamu ja *Unit of Work* mustreid. Mustrite eesmärk on luua abstraktne kiht andmete juurdepääsukihi ja rakenduse äriloogika kihi vahele. Mustrite rakendamine rakenduses aitab isoleerida andmesalves toimuvate muudatusi ning pikas perspektiivis hõlpsustab mustri intergreerimine testipõhise arendamise juurutamist [37].

6.1.4 Esitlus

Rakenduse esitluskiht koosneb otspunktidest, mis on jagatud rollipõhiseks kasutamiseks. Otspunktide visualiseerimiseks ning kaardistamiseks on kasutatud rakenduses Swagger teeki [38]. Swagger võimaldab kirjeldada API-de struktuuri, ning loob automaatselt liidese, kus on võimalik API otspunkte kasutada.

Authorize 

- Account ▼
- Document ▼
- DocumentManagement ▼
- Forms ▲
- POST /api/v{version}/Personnel/Forms/AddNewForm ▼ 
- GET /api/v{version}/Personnel/Forms/RetrieveAllAvailableForms ▼ 
- SubjectSubTypes ▼
- Schemas ▲

App.Public.DTO.v1.AttachmentDto >

App.Public.DTO.v1.CommentDto >

App.Public.DTO.v1.DataManagement.Input.FormFieldInputDto >

App.Public.DTO.v1.DataManagement.Input.FormInputDto >

Joonis 7. Swagger' vaade API otspunktidest

6.2 Kinnipeetavate portaali ja vanglate portaali arendusprotsess

Rakendused on kirjutatud kasutades Javascripti raamistikku React. Javascripti komplekteerimiseks on rakendustes kasutusel Vite. Vite kiirendab ning lihtsustab Javascripti komplekteerimist [39]. Süsteemi analüüsile tuginedes on koheselt rakendustesse integreeritud TypeScript'i funktsionaalsus. Rakenduste oleku salvestamiseks on kasutusel teek Zustand. Zustand on väike, kiire ja skaleeritav olekuhalduslahendus. Zustandil on kasutusel React'i konksudel põhinev API [39]. Rakendustes oleku salvestamine on oluline autentimise integreerimiseks. Kujundamisel on rakendustes kasutusel Bootstrap. Bootstrap on tasuta avatud lähtekoodiga CSS-i raamistik. Raamistik sisaldab veebirakendustes kasutatavate komponentide kujundusmalle [41]. Masinliidesega suhtlemiseks kasutatakse teeki Axios. Axios on JavaScript'i HTTP-klient ning võimaldab teha brauseris HTTP-päringuid ning käsitleda päringu- ja vastuseandmete teisendamist [42].

6.2.1 Kinnipeetavate portaali vaated

Kinnipeetavate portaali funktsionaalsuse täitmiseks on planeeritud kaks vaadet. Lisaks on planeeritud esileht.

Esilehe peamine eesmärk on tutvustada portaalis võimaldatavaid lahendusi.



Joonis 8. Kinnipeetavate portaali esileht

Pöördumise esitamiseks on loodud eraldi vaade. Vaate kuvamisel päritakse esmalt serverilt aktiivsed vormid, mida kinnipeetaval on võimalik täita. Vormid on andmebaasis salvestatud koos väljadega, mis võimaldavad need kasutajaliideses konstrueerida.

Loo uus pöördumine

Siin on sul võimalik luua uus pöördumine.

Vaie

Siin on lühike tutvustav.

Tallinna Vangla

Üksus I

Haldusorgani nimetus, kellele vaie esitatakse:

Vaidlustava haldusakti või toimingu sisu:

Vaide esitamise põhjused:

Vaide esitaja taotlus:

Vaide esitaja kinnitus selle kohta, et vaieldavas asjas ei ole jõustunudkohtuotsust ega toimu kohtumenetlust:

Vaidele lisatud dokumentide loetelu

Esita

Joonis 9. Kinnipeetavate portaali pöördumise loomise vaade


Esitatud pöördumiste staatustega ning vastustega tutvumiseks on loodud kolmas vaade.

>>	Pealkiri	Esitamise aeg	Pöördumise nr	Staatust
∨	Kinnipeetava taotlus	2023-05-10T07:15:19.760916Z	638193105197597797	Menetluses
∧	Kinnipeetava taotlus	2023-05-10T07:16:10.987286Z	638193105709861634	Vastatud
<p>Taotlus: : Palun lisada mind hommikusöögi nimekirja veganina!</p> <p>Vastus:Kinnipeetav lisatakse hommikusöögi nimekirja veganina VASTUS SOBIB, LÕPETAN PÖÖRDUMISE VAIDLUSTAN</p>				
∨	Kinnipeetava taotlus	2023-05-10T07:14:33.430659Z	638193104733654927	Lõpetatud
∨	Kinnipeetava taotlus	2023-05-10T07:14:54.722709Z	638193104947216816	Menetluses
∧	Kinnipeetava taotlus	2023-05-10T07:15:23.570586Z	638193105235694924	Saadetud
<p>Taotlus: : At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praestitiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.</p> <p>KONTAKT SELGITAS JA KIRJALIKKU VASTUST POLE VAJA</p>				

Joonis 10. Kinnipeetavate portaali esitatud pöördumiste vaade

6.2.2 Vanglate portaali vaated

Vanglate portaali on planeeritud lisaks esilehele veel dokumendihalduse vaade, kus on võimalik tutvuda kinnipeetavate poolt esitatud pöördumiste ning vastaja vaade, kus kasutaja saab anda vastuse pöördumisele.


Väiju

Vali asutus ja üksus

Tallinna Vangla ∨

Üksus I ∨

VALI

Joonis 11. Vanglate portaali esileht

Dokumendihalduse vaatest on võimalik minna dokumendi välju muutma või parandama, dokumendi liiki ja alamliiki määrama ning dokumendi täitjat ja kaastäitjat määrama.

The screenshot displays a document management interface. At the top, it shows '1 of 2 row(s) selected'. Below this is a table with columns: 'Select', 'Vangi eesnimi', 'Vangi perekonnanimi', 'Vangi isikukood', and 'Dokumendi loomise kp'. The first row is selected and shows 'Steve', 'Vang', '5123123123', and '2023-04-22T14:46:15.87445Z'. Below the table, there is a detailed view of the document content, including sections like 'Haldusorgani nimetus, kellele vaie esitatakse', 'Vaidlustava haldusakti või toimingu sisu', 'Vaide esitamise põhjused', 'Vaide esitaja taotlus', and 'Vaide esitaja kinnitus selle kohta, et vaieldavas asjades ei ole jõustunudkohtuotsust ega toimu kohtumenetlust'. At the bottom of the table view, there are buttons for 'REDIGEERI', 'SUUNA TAGASI', and 'REGISTREERI', along with 'Rows per page 10' and '1-2 of 2'. Below the table, there are four columns of links, each labeled 'LINKS' and containing four blue links: 'Link 1', 'Link 2', 'Link 3', and 'Link 4'.

Joonis 12. Vanglate portaali dokumendihalduse vaade

Haldusorgani nimetus, kellele vaie esitatakse:

Justiitsministeerium

Vaidlustava haldusakti või toimingu sisu:

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore

Vaide esitamise põhjused:

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur,

Vaide esitaja taotlus:

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia

Vaide esitaja kinnitus selle kohta, et vaieldavas asjad ei ole jõustunudkohtuotsust ega toimu kohtumenetlust:

Vaidele lisatud dokumentide loetelu

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur,

Määra täitja ja kaastäitja

Täitja

Megan Vastaja

Kaas täitja

Määra liik ja alamliik

Liik

Taotlus

Lattu paigutamine, Laost väljastamine

Lattu paigutamine

Laost väljastamine

Riiete pesemine

Raha kanded

Joonis 13. Vanglate portaali pöördumise registreerimine

Lisaks on planeeritud vastaja vaade, kuhu kuvatakse kasutajale vastamiseks määratud dokumendid. Vastamise vaates on võimalik anda kommentaar või vastus. Kui kasutaja on lõplik täitja, siis selles vaates koostab kasutaja lõpliku vastuse.

Haldusorgani nimetus, kellele vaie esitatakse: Justiitsministeerium

Vaidlustava haldusakti või toimingu sisu: Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

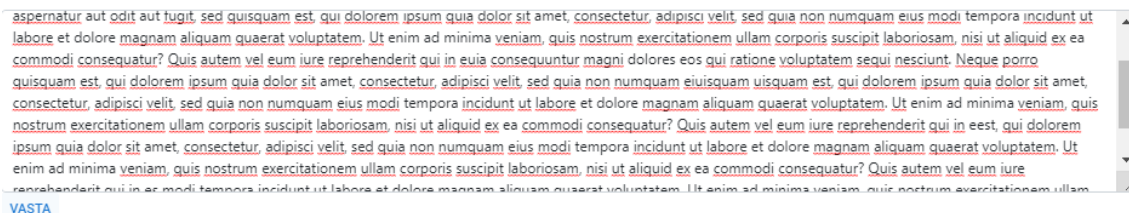
Vaie esitamise põhjused: Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

Vaie esitaja taotlus: Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

Vaie esitaja kinnitus selle kohta, et vaieldavas asjad ei ole jõustunudkohtuotsust ega toimu kohtumenetlust: on

Vaidele lisatud dokumentide loetelu: Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

Vastus pöördumisele



aspernatur aut odit aut fugit, sed quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur? Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur? Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

VASTA

Joonis 14. Vastuse andmine pöördumisele

6.3 Tulemused

Töö tulemusena koostati rakenduse realiseerimiseks esmalt masinliides, kuhu integreeriti esmane abstraktne baaskiht, mida funktsionaalsuse elluviimiseks kasutatakse. Loodi andmebaas ning andmebaasi andmemudelid. Viimaks loodi otspunktid ning otspunktide funktsionaalsuse loomiseks ülejäänud vajaminevad komponendid.

Järgnevalt loodi kasutajaliidesed mõlemale rakendusele. Kasutajaliidesesse loodi vajaminev infrastruktuur rakenduse loomiseks ning seejärel loodi ühendused masinliidesega suhtlemiseks. Viimase etapini integreeriti lõplik funktsionaalsus kasutajaliidesesse.

Töö käigus integreeriti suurem osa esialgu esitatud nõuetest ning tõestati probleemi lahendatavus.

7 Kokkuvõte

Diplomitöö eesmärgiks oli luua kinnipidamisasutustele prototüüplahendus, mis võimaldaks kinnipeetavatel elektrooniliselt kinnipidamisasutustes pöördumisi esitada ja pöördumiste vastustega tutvuda. Lahenduse realiseerimiseks oli vaja arendada kinnipeetavate poolne esiliides rakendusele, vanglate poole esiliides rakendusele ning masinliides portaalide vaheliseks suhtluseks.

Prototüübi ärianalüüsi käigus kasutati erinevaid meetodikaid, et välja selgitada komponentide funktsionaalsed nõuded. Määratleti mitte-funktsionaalsed nõuded ning kaardistati praegune ja eeldatav põhiprotsess. Prototüübi süsteemianalüüsi käigus võrreldi erinevaid tehnoloogiasid, mis olid vajalikud rakenduse realiseerimiseks ning langetati analüüsi järgselt põhjendatud otsused. Analüüsiti erinevaid võimalusi rakenduse eesmärkide täitmiseks. Samuti langetati arhitektuurilised otsused, mis võimaldasid probleemi lahendada. Diplomitöö praktilises osas demonstreeriti lähenemisviisi käesolevale probleemile koos illustreerivate näidetega, näitamaks kuidas kasutajaliides rakendustes saab olema.

Loodud prototüüp katab suurema osa analüüsi käigus määratletud nõuetest ning rakendused on diplomitöös sõnastatud eeldustega kooskõlas.

Kasutatud kirjandus

- [1] „Karjäär ja töökohad. Vabad töö ja praktikakohad,“ [Võrgumaterjal]. Available: <https://www.vangla.ee/et/karjaar-ja-tookohad/vabad-too-ja-praktikakohad>. [Kasutatud 2 April 2023].
- [2] S. j. R.Sults, „Pöördumise määratlemine,“ 19 09 2007. [Võrgumaterjal]. Available: http://stud.sisekaitse.ee/Pollumae/VangOI_kinnipavaldu/1_prdumise_mratilem.html. [Kasutatud 2 04 2023].
- [3] „Karistuse kandmine/Kaebused ja vastuvõtt/Vangi kaebeõigus ja -võimalused,“ [Võrgumaterjal]. Available: <https://www.vangla.ee/et/kaebused-ja-vastuvott/vangi-kaebeoigus-ja-voimalused>. [Kasutatud 02 04 2023].
- [4] S. j. R.Sults, „Esileht,“ 19 09 2007. [Võrgumaterjal]. Available: http://stud.sisekaitse.ee/Pollumae/VangOI_kinnipavaldu/index.html. [Kasutatud 02 04 2023].
- [5] „Vangla ja kriminaalhooldus,“ [Võrgumaterjal]. Available: <https://www.just.ee/kuritegevus-ja-selle-ennetus/vangla-ja-kriminaalhooldus>. [Kasutatud 02 04 2023].
- [6] „Riigihanke „Kinnipeetavate isikute portaali tellimine“ arendushanke tehniline kirjeldus,“ RIK, Tallinn, 2022.
- [7] Tartu Vangla, *Tartu vangla kodukord*, Tartu, 2013.
- [8] Riigi Infosüsteemi Amet, „DOKUMENDIHALDUSSÜSTEEM DELTA,“ [Võrgumaterjal]. Available: <https://www.riha.ee/Infos%C3%BCsteemid/Vaata/ria-delta>. [Kasutatud 05 04 2023].
- [9] S. j. R.Sults, „Pöördumise kontroll,“ 19 09 2007. [Võrgumaterjal]. Available: http://stud.sisekaitse.ee/Pollumae/VangOI_kinnipavaldu/4_prdumise_kontroll.html. [Kasutatud 04 04 2023].
- [10] D. P. P. T. James Cadle, *Business Analysis Techniques*, British Informatics Society Limited, 2010.
- [11] J. Patton, *User Story Mapping*, O'Reilly Media, 2014.
- [12] J. Bloch, „How to design a good API and why it matters,“ Oregon, 2006.
- [13] A. K. Keith Harman, „Learning Content Management System,“ %1 *Learning Objects: Standards, Metadata, Repositories, and LCMS*, California, Informing Science Press, 2007, p. 157.
- [14] Forumone, „The Benefits of a Flat-File CMS: Management, Performance, Security,“ Forumone, 16 08 2019. [Võrgumaterjal]. Available: <https://www.forumone.com/insights/blog/the-benefits-of-a-flat-file-cms-management-performance-security/#:~:text=A%20flat%2Dfile%20CMS%20is,folders%20rather%20than%20a%20database>. [Kasutatud 07 04 2023].
- [15] J. Hallahan, Interviewee, *Harnessing the power of React in design prototyping*. [Intervjuu]. 27 03 2018.
- [16] „What is TypeScript?,“ Microsoft, [Võrgumaterjal]. Available: <https://www.typescriptlang.org/>. [Kasutatud 21 04 2023].

- [17] A. S. Mariana Berga, „imaginarycloud,“ 15 03 2023. [Võrgumaterjal]. Available: <https://www.imaginarycloud.com/blog/grpc-vs-rest/>. [Kasutatud 13 04 2023].
- [18] „protobuf,“ [Võrgumaterjal]. Available: <https://protobuf.dev/overview/>. [Kasutatud 15 04 2023].
- [19] Indeed Editorial Team, „Indeed,“ 20 12 2022. [Võrgumaterjal]. Available: <https://www.indeed.com/career-advice/career-development/what-is-net#:~:text=Improved%20productivity-,The%20.,quality%20applications%20in%20less%20time.> [Kasutatud 15 04 2023].
- [20] „Why Node.js Is Ideal for Creating Backend APIs,“ 11 02 2022. [Võrgumaterjal]. Available: <https://centogram.com/2022/02/11/why-node-js-is-ideal-for-creating-backend-apis/>. [Kasutatud 16 04 2023].
- [21] G. Singhal, „logrocket,“ 03 03 2022. [Võrgumaterjal]. Available: <https://blog.logrocket.com/django-rest-framework-create-api/#:~:text=HTML%2C%20and%20more.-,Why%20Django%20REST%20framework%3F,you%27re%20familiar%20with%20Django.> [Kasutatud 17 04 2023].
- [22] [Võrgumaterjal]. Available: <https://swagger.io/docs/specification/2-0/what-is-swagger/>. [Kasutatud 22 04 2023].
- [23] M. Naik, „Database Management System Performance Analysis and Comparison,“ 2011. [Võrgumaterjal]. Available: <https://scholarworks.calstate.edu/downloads/qr46r094j>. [Kasutatud 16 04 2023].
- [24] „What is a relational database?,“ IBM, [Võrgumaterjal]. Available: <https://www.ibm.com/topics/relational-databases>. [Kasutatud 2023 04 16].
- [25] K. D. Foote, „NoSQL Databases: Advantages and Disadvantages,“ 17 11 2022. [Võrgumaterjal]. Available: <https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/>. [Kasutatud 2023 04 19].
- [26] S. R. Montaña, „NoSQL for Artificial Intelligence,“ 20 09 2018. [Võrgumaterjal]. Available: <https://www.slideshare.net/SebastinRamrezMontao/nosql-for-artificial-intelligence>. [Kasutatud 20 04 2023].
- [27] PostgreSQL Global Development Group, „About,“ [Võrgumaterjal]. Available: <https://www.postgresql.org/about/>. [Kasutatud 2023 04 20].
- [28] M. Burggraf, „SQL or NoSQL? Why not use both (with PostgreSQL)?,“ 24 11 2022. [Võrgumaterjal]. Available: <https://supabase.com/blog/sql-or-nosql-both-with-postgresql>. [Kasutatud 21 04 2023].
- [29] „Introduction to JSON Web Tokens,“ [Võrgumaterjal]. Available: <https://jwt.io/introduction>. [Kasutatud 21 04 2023].
- [30] Microsoft, „What's new in .NET 7,“ 03 09 2023. [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/dotnet/core/whats-new/dotnet-7>. [Kasutatud 22 04 2023].
- [31] „Use Docker Compose,“ [Võrgumaterjal]. Available: https://docs.docker.com/get-started/08_using_compose/#:~:text=Docker%20Compose%20is%20a%20tool,or%20tear%20it%20all%20down. [Kasutatud 22 04 2023].
- [32] Microsoft, „Entity Framework Core,“ 25 05 2021. [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/ef/core/>. [Kasutatud 22 04 2023].

- [33] Microsoft, „Managing Database Schemas,“ 09 09 2020. [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/ef/core/managing-schemas/>. [Kasutatud 22 04 2023].
- [34] Microsoft, „Migrations Overview,“ 01 12 2023. [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/ef/core/managing-schemas/migrations/?tabs=dotnet-core-cli>. [Kasutatud 22 04 2023].
- [35] L. W. S. L. O. Q. Yihan Duan, „Layered Architecture,“ 22 07 2019. [Võrgumaterjal]. Available: https://cs.uwaterloo.ca/~m2nagapp/courses/CS446/1195/Arch_Design_Activity/Layered.pdf. [Kasutatud 22 04 2023].
- [36] „What is AutoMapper?,“ [Võrgumaterjal]. Available: <https://automapper.org/>. [Kasutatud 22 04 2023].
- [37] Microsoft, „Implementing the Repository and Unit of Work Patterns in an ASP.NET MVC Application,“ 07 01 2022. [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/aspnet/mvc/overview/older-versions/getting-started-with-ef-5-using-mvc-4/implementing-the-repository-and-unit-of-work-patterns-in-an-asp-net-mvc-application>. [Kasutatud 22 04 2023].
- [38] „What is Swagger?,“ [Võrgumaterjal]. Available: <https://swagger.io/docs/specification/2-0/what-is-swagger/>. [Kasutatud 23 04 2023].
- [39] „Why Vite,“ [Võrgumaterjal]. Available: <https://vitejs.dev/guide/why.html>. [Kasutatud 23 04 2023].
- [40] „Introduction,“ [Võrgumaterjal]. Available: <https://docs.pmnd.rs/zustand/getting-started/introduction>. [Kasutatud 23 04 2023].
- [41] „Get started with Bootstrap,“ [Võrgumaterjal]. Available: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>. [Kasutatud 23 04 2023].
- [42] „What is Axios?,“ [Võrgumaterjal]. Available: <https://codebots.com/docs/what-is-axios#:~:text=Axios%20is%20a%20promised%2Dbased,of%20request%20and%20response%20data..> [Kasutatud 23 04 2023].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Karl Jensen

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Kinnipeetavate apellatsiooniportaali prototüüp" , mille juhendaja olid Jaanus Pöial ja Mark-Hendrik Mäeste
 - 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.

15.04.2023

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 punktile 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.