TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Hugo Valk 193834IABB

Biomeetrilise isikutuvastuse abil noorukite Smart-ID registreerimisprotsessi loomine

Bakalaureusetöö

Juhendaja: Tarvo Treier
MSc

Tallinn 2022

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: Hugo Valk

08.05.2022

Annotatsioon

Käesoleva bakalaureusetöö eesmärgiks on luua biomeetrilise isikutuvastuse abil noorukite Smart-ID registreerimisprotsess.

Eesmärgi saavutamiseks uuriti olemasolevaid noorukitele loodud Smart-ID registreerimisprotsesse ja täiskasvanutele loodud biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessi. Seejärel tutvus autor olemasolevate nõuetega ja analüüsis, kuidas saaks olemasolevaid protsesse noorukite biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessi loomiseks kasutada. Analüüsi tulemusel selgus, et olemasolevaid Smart-ID registreerimisprotsesside samme saab ära kasutada noorukite biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessi loomiseks.

Tulemina valmis serveripoolne liides, mida SK ID Solutions AS kasutab noorukite biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessi jaoks. Loodud lahendus aitab noorukitel registreerida Smart-ID konto lihtsamalt, hoides kokku raha ja aega. Tulemuste analüüsist selgus, et loodud protsessi äriline kasu väljendub peamiselt Smart-ID tehingute arvu kasvus ja protsessi biomeetrilise isikutuvastuse sammu saab ära kasutada ka mobiil-ID konto registreerimisprotsessis või teistes protsessides.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 24 leheküljel, kuut peatükki, kaheksat joonist, üht tabelit.

Abstract

Creating the Smart-ID registration process for minors using biometric identification

The purpose of this bachelor's thesis is to create the Smart-ID registration process for minors using biometric identification.

To achieve the goal, the existing Smart-ID registration processes for minors and the Smart-ID biometric registration process for adults were examined. Following that, the existing requirements were examined, and an analysis of how previous processes could be used to create the new process was conducted. The analysis revealed that the current steps of Smart-ID registration processes can also be used to create the Smart-ID registration process for minors using biometric identification.

The result of this thesis is a server-side interface that SK ID Solutions AS employs for the Smart-ID biometric registration process for minors. The created solution helps minors to register a Smart-ID account more easily, saving money and time. The analysis of the results revealed that the business benefits of the created process are primarily reflected in the increase in the number of Smart-ID transactions, and the biometric identification step of the process can also be used in the mobile-ID account registration process or other processes.

The thesis is in Estonian and contains 24 pages of text, six chapters, eight figures, one table.

Lühendite ja mõistete sõnastik

API	*Application Programming Interface*, liides rakenduse loogikale
CSR	*Certificate Signing Request*, sertifikaadi allkirjastamise taotlus
Eestkostja	Nooruki seaduslik esindaja või hooldaja
eIDAS	*Electronic IDentification, Authentication and trust Services*, Euroopa Liidus kehtiv e-identimise ja e-tehingute määrus
HTTP	*HyperText Transfer Protocol*, sõnumipõhine keel ja protokoll, mis aitab arvutitel üle veebi suhelda
ID-kaart	Eesti kodaniku ja Eestis püsivalt elava Euroopa Liidu kodaniku jaoks kohustuslik isikut tõendav dokument, mida väljastab Politsei- ja Piirivalveamet [1]
IProov	Näotuvastusteenuseid pakkuv ettevõte
JDBC	*Java Database Connectivity*
JPA	*Java Persistence API*
JVM	*Java Virtual Machine*
Koodi refaktoreerimine	Koodi struktuuri parandamine, nii et selle funktsionaalsus ei muutu
Mobiil-ID	SIM-kaardi põhine elektrooniline isikutuvastusteenus, mis võimaldab kliendil end elektrooniliselt tuvastada ja anda elektroonilisi allkirju [2]
MRZ	*Machine-Readable Zone*, masinloetav ala
NFC	*Near Field Communication*, lähiväljaside
Nooruk	Alla 18-aastane isik
OCR	*Optical Character Recognition*, optiline märgituvastus
PKI	*Public Key Infrastructure*, avaliku võtme taristu
QES	*Qualified Electronic Signature*, e-allkirja kõrgeim tase, mis on võrdsustatud omakäelise allkirjaga
QSCD	*Qualified Signature Creation Device*, kõrgeima võimaliku usaldustasemega elektroonilise allkirjastamise teenus
ReadID	Mobiiltelefoni abil isikut tõendavate dokumentide usaldusväärsust kontrolliv ettevõte
REST	*Representational State Transfer*, levinud tarkvaraarhitektuuri stiil veebiteenuste loomiseks

SK API	SK ID Solutions AS serveripoolne liides rakenduse loogikale
Smart-ID	Seadmepõhine elektrooniline isikutuvastusteenus, mis võimaldab kliendil end elektrooniliselt tuvastada ja anda elektroonilisi allkirju [2]

Sisukord

1 Sissejuhatus 11
1.1 Eesmärk 11
1.2 Ülevaade tööst 12
2 Taust 13
2.1 Smart-ID 13
2.2 Smart-ID võrdlus mobiil-ID-ga 14
2.3 Biomeetriline isikutuvastus 14
2.4 Biomeetriline isikutuvastus Smart-ID registreerimisprotsessis 15
3 Analüüs 16
3.1 Olemasolevad noorukite Smart-ID registreerimisprotsessid 16
3.1.1 Kasutades elektroonilist isikutuvastust ID-kaardiga 16
3.1.2 Külastades pangakontorit koos eestkostjaga 17
3.2 Olemasolev täiskasvanute biomeetrilise isikutuvastusega Smart-ID registreerimisprotsess 19
3.3 Noorukite biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessi analüüs, kasutades olemasolevaid protsesse 21
3.4 Dokumentide digiallkirjastamise teenusepakkujate analüüs 22
3.5 Noorukite biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessi analüüsi tulemus 23
4 Realisatsioon 25
4.1 Tehnoloogiad 25
4.2 Arendusprotsess 25
4.3 Registreerimisprotsessi sammud 26
4.4 Parameetrilise polümorfismi juurutamine 27
4.5 Testimine 28
4.5.1 Ühiktestimine 28
4.5.2 Integratsioonitestimine 29
5 Tulemuste analüüs 30
5.1 Äriline kasu 30

5.2 Eneseanalüüs 31
5.3 Tulevikulahendusi 31
6 Kokkuvõte 33
Kasutatud kirjandus 35
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks 37
Lisa 2 – SessionService klass 38
Lisa 3 – AbstractSessionService klass 39
Lisa 4 – AdultSessionService klass 40
Lisa 5 – MinorSessionService klass 41

Jooniste loetelu

Joonis 1. Noorukite Smart-ID registreerimisprotsess kasutades ID-kaarti..... 17
Joonis 2. Noorukite Smart-ID registreerimisprotsess külastades pangakontorit koos eestkostjaga..... 18
Joonis 3. Täiskasvanute Smart-ID registreerimisprotsess kasutades biomeetrilist isikutuvastust. 20
Joonis 4. Smart-ID registreerimisprotsesside sarnased sammud..... 21
Joonis 5. Noorukite Smart-ID registreerimisprotsess kasutades biomeetrilist isikutuvastust. 24
Joonis 6. Parameetrilise polümorfismi juurutamine Session klasside näitel. 28
Joonis 7. SessionDao klassi näide kasutades JPA programmeerimisliidest. 32
Joonis 8. SessionDao klassi näide kasutades JDBC programmeerimisliidest. 32

Tabelite loetelu

Tabel 1. Smart-ID võrdlus mobiil-ID-ga [2], [8]..... 14

1 Sissejuhatus

Smart-ID kasutamine isikutuvastuseks või dokumentide allkirjastamiseks on tänu oma lihtsusele ja turvalisusele iga aastaga järjest rohkem populaarsust kogunud. 10.04.2022 seisuga kasutab Balti riikides rakendust 3 196 555 kasutajat [3]. Hetkel on probleemiks liigsed kulud pankades ja komplitseeritud Smart-ID konto registreerimine noorukitele. Noorukitele on loodud kolm meetodit Smart-ID konto loomiseks: kasutades ID-kaarti, mobiil-ID-d või külastades pangakontorit. Kuna paljudel noorukitel pole ID-kaarti ega mobiil-ID-d, siis pole neil võimalik elektroonilisel viisil kontot luua, mistõttu on ainuke võimalus külastada pangakontorit. Selle tõttu peavad pangad rohkem tellereid Smart-ID registreerimisprotsessi toimimiseks tööle rakendama, kuna see võtab aega ja isiku tuvastamiseks on kohustuslik nelja silma printsiip ehk kaks tellerit peab nooruki isiku tuvastama. Nooruki jaoks on probleemiks aja- ja rahakulu, kuna pangakontorisse minemiseks on vaja aega, protsess on ajakulukas ja konto loomine on tasuline.

1.1 Eesmärk

Bakalaureusetöö eesmärgiks on realiseerida Smart-ID biomeetrilise isikutuvastusega registreerimisprotsess noorukitele, mis võimaldab isikutuvastuse jaoks kasutada kiibiga isikut tõendavat dokumenti koos näotuvastusega.

Eesmärgi saavutamiseks on autor planeerinud täita järgnevad alameesmärgid:

- Analüüsida olemasolevaid noorukitele loodud Smart-ID registreerimisprotsesse;
- Analüüsida olemasolevat täiskasvanutele loodud Smart-ID biomeetrilise isikutuvastuse registreerimisprotsessi;
- Tutvuda olemasolevate nõuetega ja analüüsida, kuidas saaks olemasolevaid protsesse uue protsessi loomiseks kasutada;
- Arendada ja testida nõuetele vastav kood.

1.2 Ülevaade tööst

Bakalaureusetöö koosneb sissejuhatusest, taustast, kolmest sisulisest peatükist ja kokkuvõttest. Teises peatükis antakse ülevaade Smart-ID-st, biomeetrilisest isikutuvastusest ja biomeetrilisest isikutuvastusest Smart-ID registreerimisprotsessis. Kolmandas peatükis analüüsitakse olemasolevaid noorukite ja täiskasvanute Smart-ID registreerimisprotsesse ning dokumentide digiallkirjastamise teenusepakkujaid. Analüüs viidi läbi, et saada ülevaade, mida saaks varasematest Smart-ID protsessidest uue registreerimisprotsessi loomiseks kasutada ja mida tuleks teisti teha. Neljandas peatükis kirjeldatakse lahenduse realisatsiooni, kus antakse ülevaade kasutatud tehnoloogiatest, arendusprotsessist, registreerimisprotsessi jaoks loodud sammudest, kasutatuimast arendustehnikast ja testimisest. Viiendas peatükis tehakse tulemuste analüüs. Tuuakse välja äriline kasu, viiakse läbi eneseanalüüs ja pakutakse välja lahendusi edaspidiseks. Viimases peatükis antakse ülevaade tehtud tööst.

2 Taust

Käesolevas peatükis tutvustatakse Smart-ID-d ja võrreldakse seda mobiil-ID-ga. Tutvustatakse biomeetrilise isikutuvastuse üldist olemust ja kasutust Smart-ID registreerimisprotsessis. Eesmärgiks on anda parem ülevaade olemasolevatest tehnoloogiatest, et oleks kergem töö sisu mõista.

2.1 Smart-ID

Smart-ID on uue põlvkonna elektrooniline isikutuvastusteenus, mis võimaldab kliendil end elektrooniliselt tuvastada ja anda elektroonilisi allkirju. Smart-ID toodi Baltikumi turule 2017. aasta veebruaris SK ID Solutions AS poolt [4]. Rakendust on võimalik kasutada ilma lisaseadmeteta nutitelefonis või tahvelarvutis. Smart-ID on tunnustatud QSCD-na (*Qualified Signature Creation Device*) – see annab võimaluse anda digiallkirja, mis on võrdväärne omakäelise allkirjaga kõigis Euroopa Liidu riikides. [5]

Smart-ID kasutab kaheastmelist isikutuvastust ehk kasutajal on vaja isiklikku nutiseadet ja PIN-koode, mille kasutaja endale on loonud. Smart-ID leiab peamiselt kasutust e-teenustesse sisenemiseks, tehingute kinnitamiseks või digitaalallkirjade andmiseks. Igapäevane Smart-ID kasutamine algab isiku kasutajatunnuse sisestamisega e-teenuse veebilehel. Pärast seda kuvatakse kontrollkood, mille kasutaja peab valima enda Smart-ID rakenduses, et protsessiga jätkata. Seejärel on vaja sisestada, kas PIN1-kood või PIN2-kood vastavalt hetkel tehtavale toimingule. [6]

Smart-ID turvalisuse tagab SplitKey tehnoloogia. Tehnoloogia tugineb avaliku võtme krüptograafia, digitaalallkirja skeemide ja PKI (*Public Key Infrastructure*) põhimõtetele. Privaatvõtmete paar on jaotatud kaheks osaks, mida Smart-ID kasutab tehingute autoriseerimiseks. Üks osa asub Smart-ID serveris ja teine osa on peidetud kasutaja nutiseadme rakendusse. Nutiseadme kasutamise lõpetamisel või konto kustutamisel kaob seadmes olev Smart-ID privaatvõti jäädavalt. [7]

2.2 Smart-ID võrdlus mobiil-ID-ga

Smart-ID peamine eelis üle mobiil-ID on lihtsus ja mugavus, mis tuuakse välja järgnevas tabelis (vt Tabel 1):

Tabel 1. Smart-ID võrdlus mobiil-ID-ga [2], [8].

	Smart-ID	**Mobiil-ID**
Tehniline vahe	Sõltub nutiseadmest, kuhu rakendus paigaldatud on.	Sõltub SIM-kaardist. SIM-kaarti saab ühest seadmest teise tõsta.
Väljastaja	Rakenduse saab paigaldada igasse nutiseadmesse.	Operaatorid, kes väljastavad SIM-kaarte. Peale SIM-kaardi saamist tuleb taotleda sertifikaadid Politsei- ja Piirivalveametist.
Maksumus kasutajale	Tasuta.	Igakuine kulu ligikaudu üks euro.
Piirangud	Smart-ID väljastamisel kehtib seadustest tulenev vanusepiirang teovõimelisuse vanuse osas – alla seitsme aastased ei ole teovõimelised. Alla 18-aastase konto peab autoriseerima eestkostja.	Vähemalt 15-aastane. Alla 18-aastase konto peab autoriseerima eestkostja.
Liitumine	Liitumiseks tuleb alla laadida rakendus ja luua konto külastades pangakontorit, kasutades ID-kaarti või mobiil-ID-d. Konto uuendamiseks või uue konto loomiseks teise seadmesse saab täiskasvanu kasutada biomeetrilist isikutuvastust.	Tuleb külastada mobiilioperaatori esindust koos isikut tõendava dokumendiga.
Kehtivusaeg	Alates sertifikaatide väljastamisest kehtivusaeg kolm aastat.	Alates sertifikaatide väljastamisest kehtivusaeg viis aastat.

2.3 Biomeetriline isikutuvastus

Biomeetriline isikutuvastus on tuvastus eristavate füsioloogiliste või käitumuslike tunnuste alusel. Selle ülesandeks on isik eelnevalt kindlaks määratud identiteediga

seostada või lahutada. Paljud käitumuslikud või füsioloogilised tunnused on iga inimese puhul erinevad. Tänu sellele on biomeetrilised identifikaatorid oma olemuselt usaldusväärsed. [9], [10]

Biomeetrilise isikutuvastuse suur eelis traditsiooniliste isikutuvastusvahendite ees on mugav kasutus, kuna isikutel pole vaja paroole või PIN-koode meeles pidada. Biomeetrilised isikutuvastusviisid muutuvad järjest odavamaks ja lihtsamaks. Biomeetria on tõhus viis privaatsuse kaitsmiseks ja pettustest kõrvalehoidmiseks, seepärast kasutatakse tulevikus arvatavasti seda tehnoloogiat enamikes tehingutes, mis nõuavad isikutuvastamist või allkirjastamist. [11]

2.4 Biomeetriline isikutuvastus Smart-ID registreerimisprotsessis

Konto loomiseks või uuendamiseks biomeetria abil on kasutajal vaja kõigepealt olla kindel, et tal on olemas:

- Varasem Smart-ID konto;
- Kiibiga dokument (nutitelefoniga NFC (*Near Field Communication*) lugeja abil andmete kättesaamine);
- NFC toega nutitelefon;
- Kaameraga nutitelefon (MRZ (*Machine-Readable Zone*) pildistamine dokumendilt ja näokontroll, mis võrdleb isiku dokumendil olevat fotot kasutajaga). [5]

Esimeseks sammuks on vormi täitmine, kus sisestatakse kontaktandmed, antakse vastavad nõusolekud ja valitakse dokumenditüüp, mida konto loomiseks kasutatakse. Eestis saab biomeetriliseks isikutuvastuseks kasutada ainult passi. Lätis ja Leedus on võimalik kasutada ID-kaarti, passi või elamisloakaarti. Teiseks sammuks on dokumendiandmete lugemine, kus esmalt pildistatakse nutiseadmega dokumendi MRZ koodi. Pärast seda viiakse läbi nutiseadme NFC lugejaga dokumendi kiibi skaneerimine. Viimaseks sammuks on IProov abiga isikusamasuse kontroll, kus võrreldakse dokumendil olevat fotot isikuga.

3 Analüüs

Käesolevas peatükis uuritakse olemasolevaid noorukitele loodud Smart-ID registreerimisprotsesse ja täiskasvanutele loodud biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessi. Eesmärgiks on analüüsida, kuidas saaks olemasolevaid protsesse noorukite biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessi loomiseks kasutada. Lisaks analüüsitakse dokumentide digiallkirjastamise teenusepakkujaid. Autor võttis lihtsustatud äriprotsesside diagrammide loomisel aluseks Nortal AS-i poolt loodud diagrammid ja nõuded.

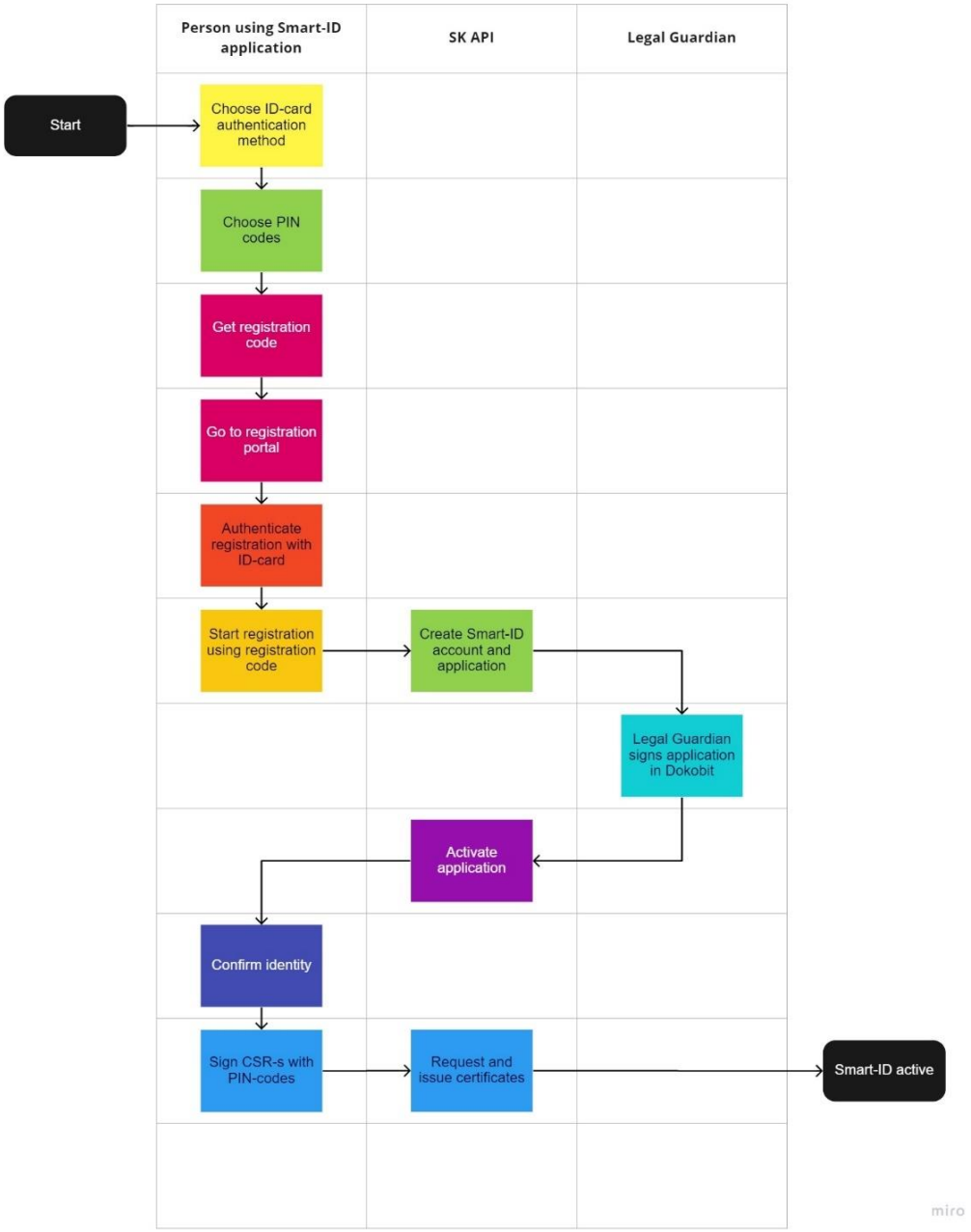
Iga protsessi kohta tuuakse välja nelja silma printsiip ehk isik peab olema kaks korda tuvastatud, et olla kindel isikusamasuses.

3.1 Olemasolevad noorukite Smart-ID registreerimisprotsessid

Biomeetrilise protsessi analüüsi jaoks alustati protsessi võrdlust varasemate noorukite Smart-ID registreerimisprotsessidega, mis ei kasuta biomeetrilist isikutuvastust.

3.1.1 Kasutades elektroonilist isikutuvastust ID-kaardiga

Registreerimisprotsess Smart-ID rakenduses algab ID-kaardi tuvastamismeetodi valimisega (vt Joonis 1). Pärast seda määrab nooruk sobivad PIN-koodid, mida tulevikus kasutada soovib. Smart-ID rakendus genereerib seejärel registreerimiskoodi, mis tuleb registreerimisportaali sisestada, et konto loomist alustada. Enne registreerimiskoodi sisestamist tuleb noorukil ennast portaalis ID-kaardiga tuvastada ja seejärel on võimalik alustada konto loomisega kasutades eelnevalt mainitud koodi. Konto loomisel teeb portaal esimese päringu SK API (SK ID Solutions AS serveripoolne liides rakenduse loogikale) poole, mis loob konto ja allkirjastamata taotluse. Taotluse allkirjastamiseks peab nooruki eestkostja Dokobit portaalis või rakenduses digiallkirja andma. Pärast allkirja andmist aktiveeritakse nooruki konto ka SK API poolel. Järgmiseks peab nooruk enda isikuandmed kinnitama ja sertifikaatide taotluse registreerimisprotsessi alguses loodud PIN-koodidega allkirjastama. Kui sertifikaadid on välja antud, saab nooruk loodud Smart-ID kontot kasutama hakata.



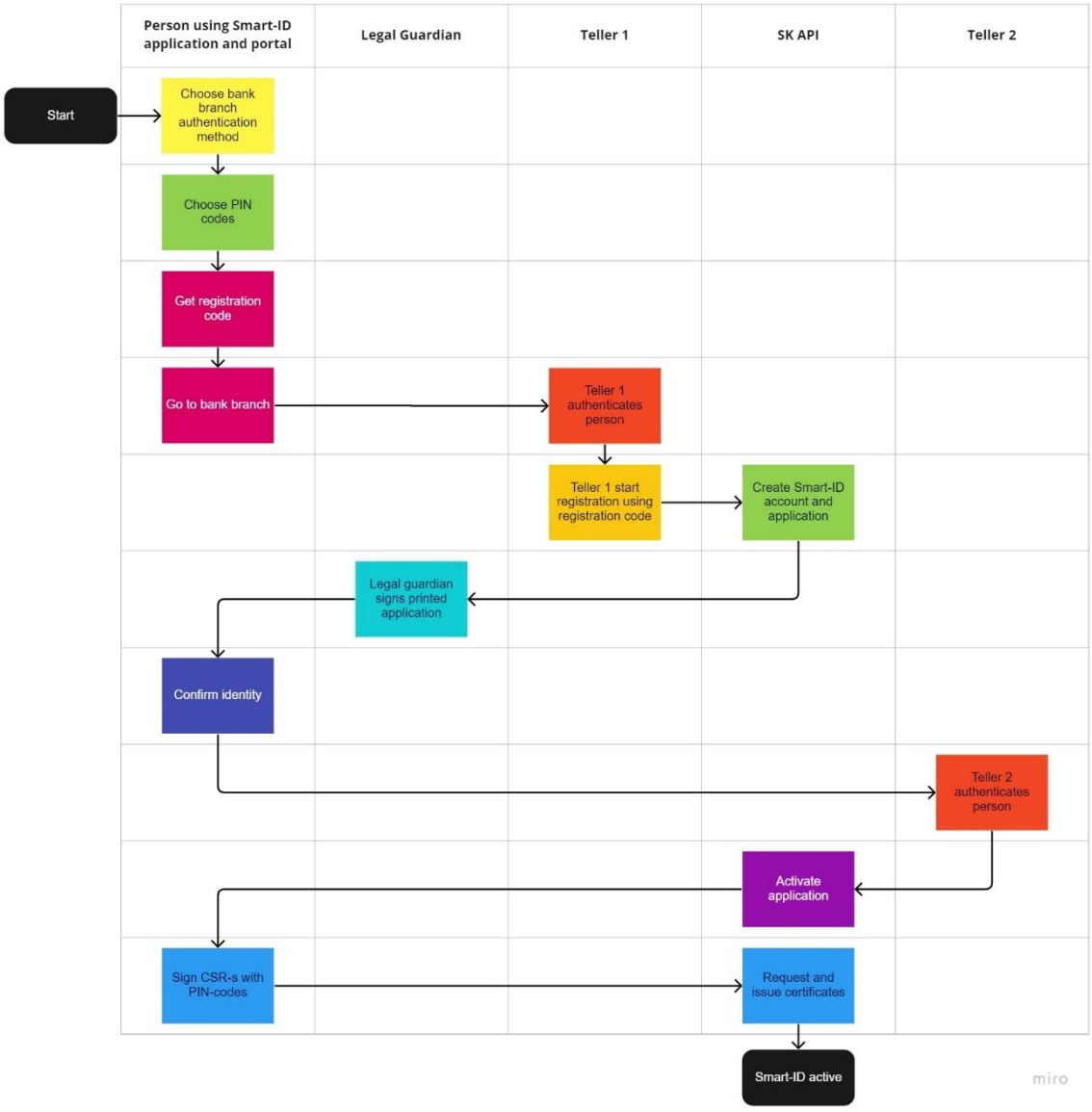
Joonis 1. Noorukite Smart-ID registreerimisprotsess kasutades ID-kaarti.

Protsessis moodustab nelja silma printsiibist ühe silmapaari ID-kaardi varasem väljastamine riigiasutuse poolt ja teise ID-kaardi PIN-koodiga tuvastamine.

3.1.2 Külastades pangakontorit koos eestkostjaga

Registreerimisprotsess algab samamoodi nagu eelnevas peatükis kuni registreerimiskoodide kättesaamiseni, mis on joonisel (vt Joonis 1) toodud välja kollase ja rohelise sammuna, erinevusega, et alguses tuleb isikutuvastuse meetodiks valida pangakontor (vt Joonis 2). Pärast registreerimiskoodide Smart-ID rakendusest kätte

saamist, tuleb noorukil koos oma eestkostjaga edasine registreerimisprotsess teostada pangakontoris. Pangakontoris on esimeseks sammuks telleri poolt isiku tuvastamine isikut tõendava dokumendiga. Seejärel sarnaneb protsess eelnevas peatükis kirjutatule, kus luuakse konto ja allkirjastamata taotlus, kuid nüüd allkirjastab tuvastatud nooruki eestkostja taotluse käsitsi. Järgmine samm on Smart-ID rakenduses nooruki poolt enda isikuandmete kinnitamine, mille järel peab teine teller samuti nooruki tuvastama. Pärast isikutuvastust jätkub protsess samamoodi nagu joonisel (vt Joonis 1), mis algab lillaga tähistatud sammust ehk konto aktiveerimisest SK API poolelt, erinevusega, et isiku andmete kinnitamise samm jääb ära, ehk tumesinisega tähistatud samm, sest see on enne teise telleri isikutuvastuse sammu juba teostatud.



Joonis 2. Noorukite Smart-ID registreerimisprotsess külastades pangakontorit koos eestkostjaga.

Pangakontoris Smart-ID registreerimisprotsessi üks suur miinus on teise telleri isikutuvastuse samm, mis on joonisel (vt Joonis 2) toodud punase kastina. Kuna pangakontoris võib teine teller tegeleda teiste tööülesannetega või hetkel mitte olemas olla, siis võib konto registreerimisprotsess nooruki jaoks ajaliselt pikeneda.

Protsessis täidetakse nelja silma printsiip nooruki tuvastamisega kahe erineva telleri poolt. Pangas registreerides ei loeta ID-kaardi väljastamist üheks silmapaariks järgnevatel põhjustel:

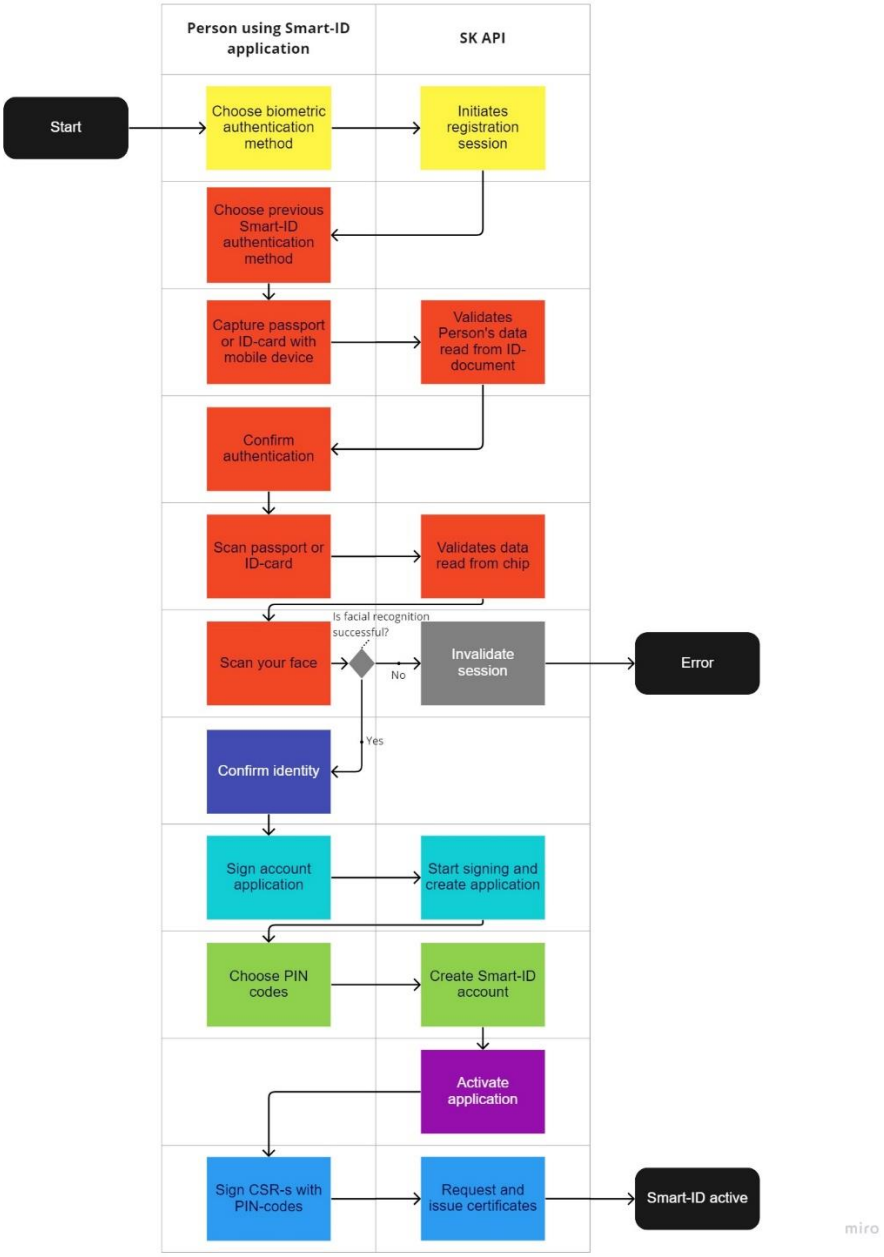
- Smart-ID loomiseks pole vajalik ID-kaardi olemasolu;
- Välisriikide isikud saavad registreeruda, seega ei saa kindel olla, kuidas teises riigis dokument väljastatud on;
- Dokumendi kontroll toimub visuaalsel teel.

3.2 Olemasolev täiskasvanute biomeetrilise isikutuvastusega Smart-ID registreerimisprotsess

Täiskasvanute biomeetrilise isikutuvastusega registreerimisprotsess on võimalik ainult kasutajatele, kellel on varasemalt Smart-ID konto olemas, seega isik, kes soovib Smart-ID-d esmakordselt luua, peab valima mõne teise olemasoleva registreerimisprotsessi. Biomeetirilise isikutuvastusega registreerimisprotsess on tehtud nii sellepärast, et täiskasvanu varasemat Smart-ID kontot kasutatakse esimeseks isikutuvastuseks. Biomeetrilist isikutuvastust on võimalik kasutada olemasoleva konto uuendamiseks samas seadmes või uue konto loomiseks nutiseadme vahetuse puhul. Autor liigub analüüsiga edasi ainult uue konto registreerimisprotsessiga, sest see sarnaneb soovitud noorukite protsessiga kõige rohkem.

Registreerimisprotsess algab Smart-ID biomeetrilise isikutuvastusmeetodi valimisega, mille järel tehakse esimene päring SK API poole, mis alustab sessiooni (vt Joonis 3). Järgmiseks tuvastatakse andmed varasemalt loodud Smart-ID kontolt. Seejärel alustatakse täiendavat isikutuvastust MRZ andmete põhjal, mis loetakse passilt või ID-kaardilt masinloetava teksti pealt ja tehakse päring SK API poole, mis valideerib isiku andmed. Pärast seda loetakse andmed isikut tõendava dokumendi NFC kiibilt ja tehakse päring SK API poole, kus võrreldakse neid MRZ andmetega. Seejärel algab rakenduses

IProov näotuvastus, kus võrreldakse isikut tõendava dokumendi peal olevat fotot ja IProov-iga tehtud videot [12]. Negatiivse näotuvastuse korral tehakse SK API poole päring, mis tunnistab sessiooni kehtetuks ja lõpetab terve protsessi veaga. Positiivse näotuvastuse korral kinnitab täiskasvanu enda isikuandmed ja algab Smart-ID allkirjastamata taotluse loomine SK API poolel, mille järel peab täiskasvanu määrama PIN-koodid, mida tulevikus Smart-ID-ga kasutama hakkab. Seejärel luuakse SK API poolel konto ja aktiveeritakse taotlus. Viimaseks sammuks on sertifikaatide taotluse allkirjastamine loodud PIN-koodidega. Pärast sertifikaatide väljastamist SK API poolt on Smart-ID konto aktiveeritud.

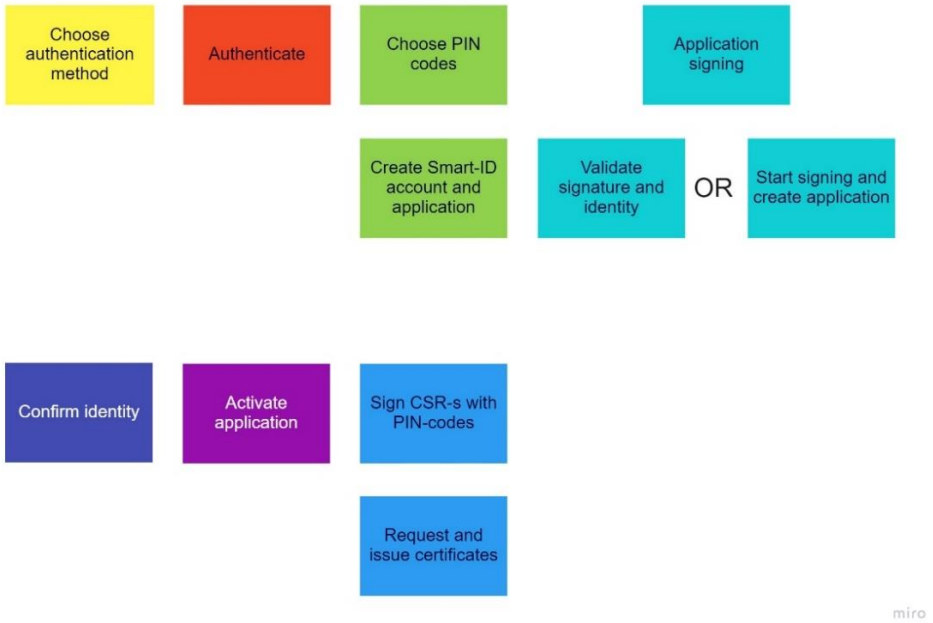


Joonis 3. Täiskasvanute Smart-ID registreerimisprotsess kasutades biomeetrilist isikutuvastust.

Protsessis moodustab nelja silma printsiibist ühe silmapaari Smart-ID konto varasem olemasolu ja teise biomeetriline isikutuvastus.

3.3 Noorukite biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessi analüüs, kasutades olemasolevaid protsesse

Kolme varasema registreerimisprotsessi võrdluse põhjal saab välja tuua, et Smart-ID konto loomine koosneb iga kord sarnastest sammudest, mis on välja toodud joonisel (vt Joonis 4), mille edukaks läbimiseks kasutatakse erinevaid sisemisi protsesse ja sammude järjekordi. Peamised lisasammud on noorukite ID-kaardi ja pangakontori külastamise registreerimisprotsessides. Nendes protsessides tuleb kasutada registreerimise alustamiseks Smart-ID rakendusest saadavat registreerimiskoodi ning seda vastavalt kas registreerimisportaali sisestama või pangakontoris tellerile näitama.



Joonis 4. Smart-ID registreerimisprotsesside sarnased sammud.

Protsesside tähtsamateks sammudeks on nelja silma printsiibi läbimine, konto loomine, konto allkirjastamata taotluse loomine ja aktiveerimine ning sertifikaatide väljastamine.

Olemasolevaid registreerimisprotsesside samme saab ära kasutada ka noorukite biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessi loomiseks. Esimese sammu jaoks saab ära kasutada varasemalt loodud täiskasvanute Smart-ID biomeertilise

isikutuvastusega registreerimisprotsessi joonisel (vt Joonis 3) toodud kollaste kastidega samme, mis tähistavad konto algatamise samme. Kuna noorukil peab olema võimalik luua konto biomeetrilise isikutuvastuse abil nii, et varasem Smart-ID konto pole vajalik, siis ei saa täiskasvanute isikutuvastusprotsessi, mis on joonisel (vt Joonis 3) toodud välja punasega enam täielikult ära kasutada. Ära tuleb jätta varasema Smart-ID konto abil isiku tuvastamise samm. Lisaks ei saa noorukite puhul tavalist biomeetrilist isikutuvastust kohe lõpuni viia, nagu on tehtud täiskasvanute registreerimisprotsessis joonisel (vt Joonis 3) punaste sammudega, sest „*Scan your face*“ sammu jaoks on vaja nooruki eestkostja nõusolekut, et protsessi alustada. Lisaks saab ära kasutada varasematest noorukite protsessidest joonistel (vt Joonis 1 ja Joonis 2) rohelisega toodud konto ja allkirjastamata taotluse loomise sammu ning türkiissinise kastiga toodud nooruki Smart-ID konto taotluse allkirjastamise sammu eestkostja poolt. Ülejäänud sammud, mida on igas varasemas protsessis kasutatud, milleks on isikuandmete kinnitamine, konto aktiveerimine ja sertifikaatide väljastamine, mis on joonisel (vt Joonis 4) teises reas tumesinise, lilla ja sinisega, saab kasutada ka noorukite Smart-ID biomeertilise isikutuvastusega registreerimisprotsessis.

3.4 Dokumentide digiallkirjastamise teenusepakkujate analüüs

Teenusepakkujate analüüs viiakse läbi, et otsustada, kas nooruki eestkostja digiallkirja andmiseks tuleks jätkata Baltikumi suurima Dokobit-iga või hakata kasutama maailmas levinud DocuSign-i. Mõlemad teenuspakkujad vastavad Smart-ID allkirjastamistaseme nõudele, milleks on QES (*Qualified Electronic Signature*). Allkirja andmisel on kontrollitud nii omaniku kui ka sertifikaadi väljaandja taust. Autori arvates on DocuSign-i üldine kasutajakogemus võrreldes Dokobit-iga märgatavalt parem, kuid Docusign-iga digiallkirja andmiseks ei saa kasutada mobiil-ID-d või Smart-ID-d, mis väljastavad QES tasemega allkirjasid Balti riikides, vaid tuleb kasutada DocuSign-i partnerit IDnow-d. See muudab protsessi ebamugavaks, kuna IDnow-d kasutades tuleks tuvastamisprotsess läbida topelt, kasutamata olemasolevat Smart-ID või mobiil-ID isikutuvastust [13]. Lisaks on Dokobit-i eelis DocuSign-i ees riigikeelne dokumentatsioon ja klienditeenindus, seadusandluse ja kliendibaasi tundmine, vajaduste parem mõistmine ning lahendus Baltimaade klientidele, sest Dokobit-i fookus on just nendel riikidel. Veel on Dokobit-i eeliseks varasem kasutus noorukite Smart-ID konto registreerimisprotsessis ID-kaardiga. Seega on mõistlik võimalikult vähe varasemat

protsessi muuta, et kasutajal oleks vähem uusi samme, tänu millele on kasutajakogemus parem. [14], [15]

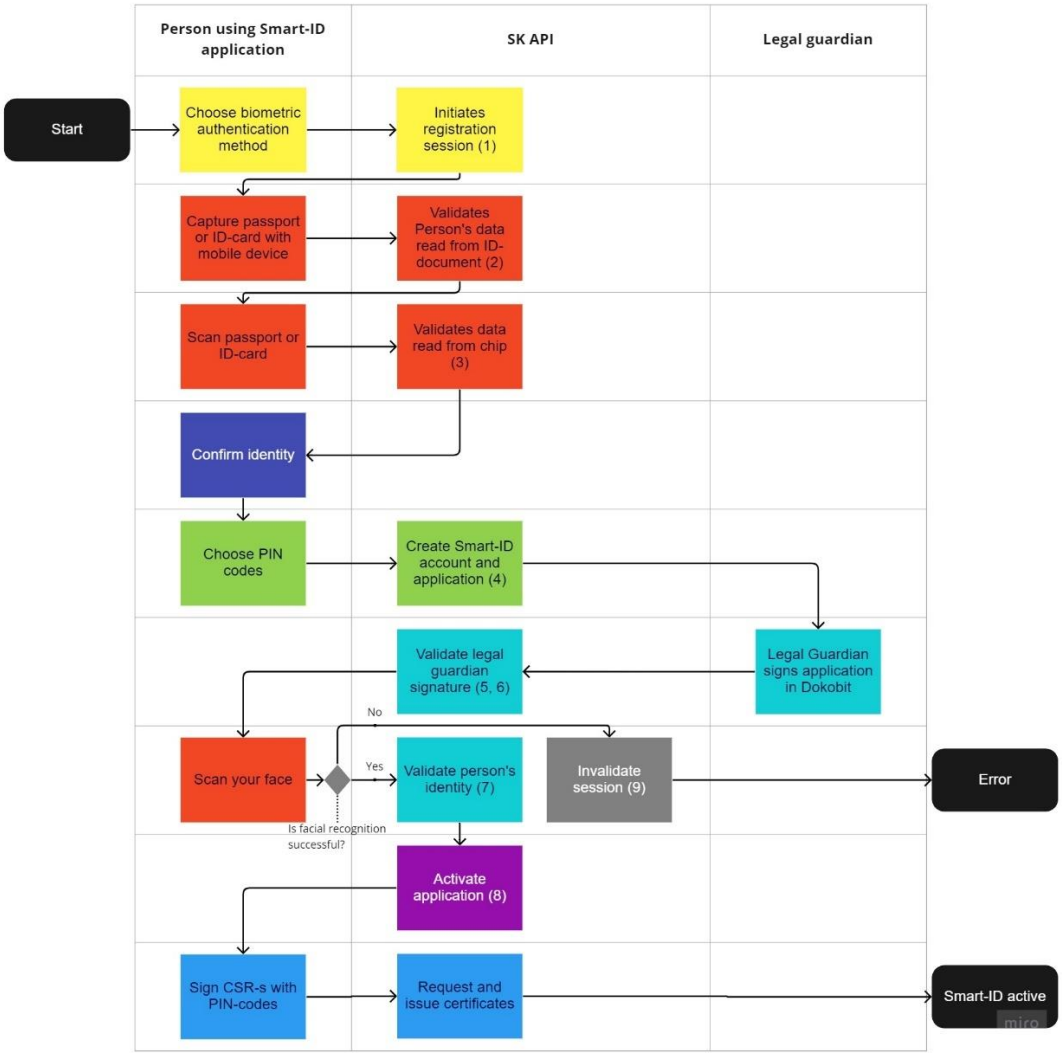
3.5 Noorukite biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessi analüüsi tulemus

Noorukite Smart-ID registreerimisprotsessi peamiseks tulevaseks eeliseks täiskasvanute Smart-ID registreerimisprotsessiga on võimalus luua Smart-ID konto kasutades biomeetrilist isikutuvastust, eelnevalt Smart-ID kontot omamata. Kuna nooruk ei saa konto taotlust ise allkirjastada, vaid seda peab tegema nooruki eestkostja, siis on varasemalt kasutatud selle jaoks ID-kaardiga registreerides digiallkirja Dokobit-is või pangas registreerides eestkostja allkirja.

Kuna digiallkirja andmise portaale on loodud mitmeid, siis tuli valida antud protsessi jaoks kõige mõistlikum ja lihtsaim lahendus. Analüüsi käigus selgus, et kõige mõistlikum oleks kasutada uuesti Dokobiti, sest see on juba hetkel laialdaselt Lätis, Leedus ja Eestis kasutusel, on klientidele mugav ning varasemalt tõestanud oma turvalisust. Peamiseks argumendiks sai ka kasutajakogemuse samaks jätmine olemasolevatele kasutajatele.

Noorukite biomeetrilise isikutuvastusega Smart-ID registreerimisprotsess algab biomeetrilise isikutuvastuse valimisega, mis käivitab SK API poolel konto algatamise sammu (vt Joonis 5). Seejärel alustatakse täiendavat isikutuvastust MRZ andmete põhjal, mis loetakse passilt või ID-kaardilt masinloetava teksti pealt ja tehakse päring SK API poole, mis valideerib isiku andmed. Pärast seda loetakse andmed isikut tõendava dokumendi NFC kiibilt ja tehakse päring SK API poole, kus võrreldakse neid MRZ andmetega. Siis kinnitab nooruk enda isikuandmed ja valib sobivad PIN-koodid, mis käivitab SK API poolel Smart-ID konto ning allkirjastamata taotluse loomise. Kui taotluse samm on edukalt läbitud, siis saab nooruki eestkostja allkirjastamise teavituse taotluse digiallkirjastamiseks. Eestkostja poolt taotluse digiallkirjastamise järel Dokobit-is kinnitatakse allkiri SK API poolel. Seejärel saab alustada isikutuvastuse viimast „*Scan your face*“ sammu, mida ei tohtinud enne eestkostja digiallkirja andmist teha. Hakkab IProov näotuvastus, kus võrreldakse isikut tõendava dokumendi peal olevat fotot ja IProov-iga tehtud videot. Negatiivse näotuvastuse korral tehakse SK API poole päring, mis tunnistab sessiooni kehtetuks ja lõpetab terve protsessi veaga. Positiivse näotuvastuse korral tehakse nooruki identiteedi kontroll SK API poolel ja jätkatakse

varasemates protsessides korduvate sammudega, milleks on taotluse aktiveerimine SK API poolel ja sertifikaatide taotluse allkirjastamine loodud PIN-koodidega. Pärast sertifikaatide väljastamist SK API poolt on Smart-ID konto aktiveeritud.



Joonis 5. Noorukite Smart-ID registreerimisprotsess kasutades biomeetrilist isikutuvastust.

Kuna noorikute näo kuju muutub kuni täiskasvanuks saamiseni palju, eriti vanuses 0–12 aastat, siis see suurendab ebaõnnestunud biomeetrilise isikutuvastuste arvu sammus „*Scan your face*“, mis on välja toodud joonisel (vt Joonis 5). See suurendab omakorda ka ebaõnnestunud registreerimisprotsesside arvu ja ärilist riski. Seega tuleb seada noorukite biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessile vanusepiirang vahemikus 13–17 aastat, millel on tunduvalt väikesem ebaõnnestumise määr. [16]

Protsessis moodustab nelja silma printsiibi nooruki biomeetriline isikutuvastus ja eestkostja digiallkiri.

4 Realisatsioon

Käesolevas peatükis kirjeldatakse arenduse tehnoloogilisi valikuid, meeskonna arendusprotsessi ja miks võeti eesmärgiks arendada töötav kood võimalikult kiiresti. Lisaks tuuakse välja kõik loodud registreerimisprotsessi sammud, autori arendatud sammud ja suurima mõjuga koodi arendustehnika. Viimaseks kirjeldatakse, kuidas tagati koodi kvaliteet, et tulevaste muudatuste või vigase protsessi korral saaks vastavalt võimalikult sujuvalt uut koodi juurde arendada või kiiresti vigu tuvastada.

4.1 Tehnoloogiad

Biomeetrilise isikutuvastuse abil noorukite Smart-ID registreerimisprotsessi serveripoolne teenus on loodud olemasoleva rakenduse peale, mille arendamiskeeleks on Java 11, kasutades Java Spring Boot raamistiku ja Oracle andmebaasi haldussüsteemi. Rakendus kasutab REST API-t (*Representational State Transfer Application Programming Interface*), mis kasutab andmetele juurdepääsuks ja nende kasutamiseks HTTP (*HyperText Transfer Protocol*) päringuid. REST API kirjeldamiseks kasutatakse Swagger liidese keelt. Swagger võimaldab koodi automaatset dokumentatsiooni ja koodi genereerimist. Andmebaasi skeemi muudatuste rakendamiseks, haldamiseks kasutab rakendus Liquibase teeki. Rakenduse kokku ehitamiseks ja haldamiseks kasutatakse Maven tööriista.

4.2 Arendusprotsess

Autori arendusmeeskond lähenes koodi arendamisele lähtudes agiilsetest meetoditest, mis aitavad tahetud lõpptulemuse kiiremini kliendini viia koos vähemate tagasilöökidega [17]. Eesmärgiks oli viia kliendini minimaalne töötav kood, mida on võimalik kasutusse võtta ja alles pärast koodi esimest üleandmist tegeleda koodi refaktoreerimise, puhastamise ja veaolukordade parema lahendamisega [18].

Meeskond töötas kasutades arendusetappe, mis kestsid kaks nädalat. Iga arendusetapi alguses valiti ja hinnati välja ülesanded, mis meeskond endale eesmärgiks teha võttis.

Sihiks oli individuaalselt alati tegeleda korraga ainult ühe ülesandega ehk enne uue ülesande võtmist pidi vana valmis olema. Arendusetapi jooksul tehti iga päev lühikesi koosolekuid, et kõik meeskonna liikmed saaksid teada anda, mis ülesandega nad hetkel tegelevad, arutada ülesannetega seotud probleeme ja tähtsaid teemasid seoses projektidega. Lisaks toimus iganädalane koosolek kliendiga, kellega koos prooviti leida olemasolevatele ja uutele probleemidele sobivaid lahendusi. Autori iga tööülesanne algas nõuetega tutvumisest ja arusaamade kooskõlastamisest. Seejärel kirjutas autor vastavalt nõuetele töötava koodi koos ühiktestidega. Järgmiseks laadis arendaja oma koodi teistele arendajatele ülevaatuseks Bitbucket-i, kus pidid arendajad enne koodi kasutuselevõttu oma kinnituse andma. Peale kinnitust suunati kood edasi testijatele, kelle ülesandeks oli tagada arenduse kvaliteet. Vigade ilmnemisel suunas testija koodi tagasi autorile, kes viis vastavad parandused sisse ja seejärel jätkus protsess sarnaselt ülaltoodule. Viimaseks sammuks oli koodi tarnimine kliendile, mis toimus vastavalt vajadusele või etteantud kuupäevadele.

4.3 Registreerimisprotsessi sammud

Noorukite Smart-ID registreerimisprotsessi serveripoolne rakendus koosneb üheksast otspunktist, millest kaheksa on vajalikud eduka protsessi läbimiseks ja üks protsessi lõpetamiseks vea puhul. Iga otspunkt on nummerdatud analüüsi tulemusel saadud joonisel (vt Joonis 5, SK API tulp):

1. startSession – alustab Smart-ID registreerimisprotsessi sessiooni ja loob juurdepääsuloa välisele teenusele;

2. ocrDataValidation – käivitab Smart-ID registreerimise kvalifitseeritud sertifikaatide jaoks, kontrollitakse MRZ andmeid ja isiku andmeid ning kinnitatakse, et kasutatav dokument poleks tühistatud;

3. dataChecks – kontrollib eelmises sammus saadud andmeid dokumendi NFC kiibi pealt saadud andmetega;

4. registerAccount – loob taotluse ja registreerib Smart-ID konto Smart-ID Core-s;

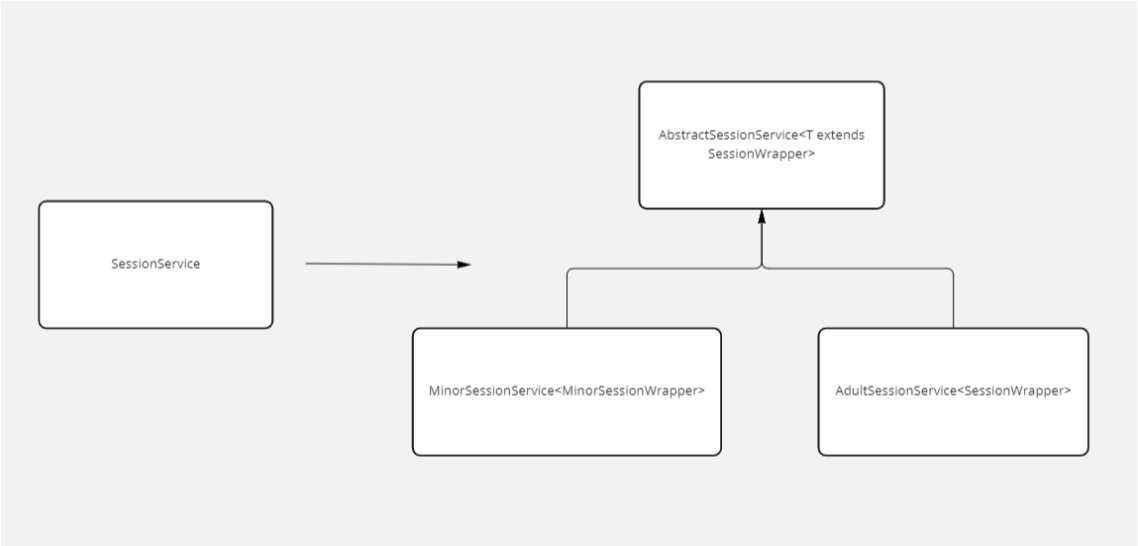
5. postback – valideerib Dokobiti allkirjastamise oleku ja valideerib nooruki eestkostja allkirja;

6. signingStatus – tagastab Smart-ID registreerimise sessiooni hetke staatuse;
7. validateIdentity – kontrollib isiku identiteedi ReadID-s ja iProov-is;
8. finalizeRegistration – muudab Smart-ID konto õigused aktiivseks Smart-ID Core-s;
9. incrementDocumentFailedSessions – muudab sessiooni kehtetuks ja suurendab dokumendi ebaõnnestunud sessioonide arvu.

Autori vastutusel Smart-ID registreerimisprotsessi serveripoolse rakenduse arendamisel olid startSession, ocrDataValidation, dataChecks, validateIdentity ja incrementDocumentFailedSessions otspunktid. Lisaks lõi autor enne arendusprotsessi algust vastava Swagger API dokumentatsiooni tervele protsessile, tegi registerAccount, postback, signingStatus ja finalizeRegistration otspunktidele koodiülevaatusi ja viis sisse parandusi. Veel refaktoreeris autor terve rakenduse kõik võimalikud klassid kahe erineva voo peale kasutades parameetrilist polümorfismi, millest on lähemalt kirjutatud järgmises peatükis.

4.4 Parameetrilise polümorfismi juurutamine

Uus noorukite Smart-ID biomeetrilise isikutuvastusega registreerimisprotsess arendati olemasolevasse rakendusse, kus oli varasemalt loodud ka täiskasvanute Smart-ID biomeetrilise isikutuvastusega registreerimisprotsess, mis on toodud välja joonisel (vt Joonis 3). Autorile tundus mõistlik jagada olemasolevad teenuse klassid, mille näide on toodud välja lisas (vt Lisa 2), kaheks erinevaks vooks, kasutades selleks parameetrilist polümorfismi, mille näide on toodud välja lisades (vt Lisa 3–5). Polümorfism aitab funktsioone ja andmetüüpe kirjutada üldisemalt, tänu sellele saab väärtusi käsitleda identselt, sõltumata tüübist [19]. Autor otsis varasematest klassidest välja ühised meetodid, mida on vaja noorukite voo jaoks ja lõi uue abstraktse klassi, mida pärib ka täiskasvanute voog (vt Joonis 6). Nooruki ja täiskasvanu teenuste klassidesse jäeti alles ainult nende klassidele vajalikud meetodid.



Joonis 6. Parameetrilise polümorfismi juurutamine Session klasside näitel.

Autor juurutas sama loogikat, mis on toodud joonisel (vt Joonis 6), ka teiste klasside ümber arendamisel. Kokku sai sama loogikat rakendada viie klassi puhul, kus sai ühe suure klassi ümber kirjutada nooruki ja täiskasvanu vooks. Pärast parameetrilise polümorfismi juurutamist tuli ka kõik nendest klassidest sõltuvad klassid üle vaadata ja õiget voogu rakendada.

4.5 Testimine

Rakenduse testimine viidi läbi kahel tasemel kasutades ühiktestimist ja integratsioonitestimist.

4.5.1 Ühiktestimine

Arendusmeeskond kirjutas ühikteste, et kontrollida iga funktsiooni töötamist ja veaolukordi. Ühiktestid on tarkvara arenduse ja juurutamise oluline samm, kuna see mitte ainult ei paranda koodi tõhusust ja töötamist, vaid vähendab ka edasise arenduse ja hoolduse regressioone [20]. Arendus viidi läbi objektorienteeritud programmeerimiskeeles Java ja testiti kõike klasside tasemel. Ühiktestimine arenduse käigus tagas autorile ja arendusmeeskonnale, et nende kirjutatud kood ja varasemalt arendatud kood on korrektne ning valmis testijatele suunamiseks.

4.5.2 Integratsioonitestimine

Integratsiooniteste kirjutasid autoriga samas arendusmeeskonnas töötavad testijad. Testide eesmärgiks on testida kogu koodi koos ehk kõikide komponentide koos töötamist. See aitab leida vigu liideses ja komponentide vahelistes toimingutes [21]. Integratsioonitestimine viidi läbi iga ülesande kohta eraldi kasutades Groovy programmeerimiskeelt, mis on JVM (*Java Virtual Machine*) põhine keel ja on kerge koos programmeerimiskeelega Java kasutada.

5 Tulemuste analüüs

Bakalaureusetöö raames valmis tarkvara, mis prognoosi kohaselt tõstab SK ID Solutions AS-i Smart-ID tehingute arvu. Antud bakalaureusetöö autor kirjutas töö koostöös ettevõttega Nortal AS, kelle kliendiks on SK ID Solutions AS. Autor oli loodava lahenduse puhul arendusmeeskonnas arendaja rollis, kuhu kuulus peale autori veel: üks tooteomanik, kaks testijat, kolm arendajat. Autor tegi seejuures arendustööd 244 tundi, mis moodustas terve meeskonna tööst ligikaudu 40%. Projekti arendamine sai alguse jaanuaris 2022. aastal ja esimene suurem tarne tehti märtsi alguses 2022. aastal. Pärast seda tehti veel väiksemaid tarneid, kas veaolukordade paremate lahenduste või täienduste jaoks.

Käesolevas peatükis kirjeldatakse SK ID Solutions AS-i ärilist kasu loodud lahendusest. Järgmiseks viib autor läbi eneseanalüüsi ja analüüsib, kuidas saaks loodud biomeetrilise isikutuvastusega noorukite Smart-ID registreerimisprotsessi ka mujal kasutada. Viimaseks toob autor olemasolevale andmebaasi programmeerimisliidesele välja mugavama alternatiivi, mida saaks tulevikus kasutusele võtta.

5.1 Äriline kasu

Peamine äriline kasu SK ID Solutions AS-ile loodud lahendusest väljendub prognoosi kohaselt tehingute arvu kasvus. E-teenuste pakkujad tasuvad iga tehingu eest vastavalt hinnakirjale [22]. Kuna enne noorukite biomeetrilise isikutuvastusega Smart-ID konto registreerimisprotsessi loomist oli konto registreerimine noorukitele komplitseeritud, siis loodud protsess teeb selle lihtsamaks. Lihtsam protsess toob omakorda juurde rohkem noorukeid, kes soovivad isikutuvastamiseks või allkirjastamiseks Smart-ID-d kasutada. Kui noorukile tekib Smart-ID kasutamise harjumus, siis jätkab ta selle kasutamist ka täisealiseks saades ja suunab suurema tõenäosusega oma pere Smart-ID-d kasutama.

Loodud lahendus suurendab konservatiivse prognoosi järgi Baltikumis biomeetrilise isikutuvastuse abil registreeritud noorukite Smart-ID kontode arvu järgmise nelja aasta jooksul igakuiselt ligikaudu kolm protsenti.

5.2 Eneseanalüüs

Lõputöö autor omandas uusi teadmisi firma üldise toimimise, meeskonnatöö, tarkvaraarenduse, dokumenteerimise ja analüüsi kohta. Kõige suurem areng toimus autoril Java programmeerimiskeele, Spring Boot raamistiku teadmiste ja tehniliste mustrite kasutamise poolel.

Autor jäi enda tehtud tööga rahule. Kui autoril oleks võimalus projekti uuesti algusest alustada, siis ta alustaks arendusprotsessiga varem ja suunaks projekti rohkem arendajaid, et ülesanded kiiremini valmis saada, sest meeskond ei suutnud esialgseks tähtajaks koodi kliendile tarnida. Veel tooks autor sisse ka üle nädala tehtavad kokkuvõtvad koosolekud, kus arutletakse tiimi eelneva kahe nädala nõrkuste ja tugevuste üle ning pannakse paika plaan, kuidas nõrkusi vähendada ja efektiivsemalt edasi töötada.

5.3 Tulevikulahendusi

Olemasolevat biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessi saaks üldiselt ära kasutada ka teistes isikutuvastamist vajavates protsessides. Selle jaoks tuleks luua eraldi rakendus, mille ainukeseks ülesandeks oleks biomeetriline isikutuvastus. Näiteks saaks seda kasutada mobiil-ID konto registreerimisprotsessi jaoks. Kui hetkel tuleb külastada mobiilioperaatori esindust koos isikut tõendava dokumendiga, siis saaks kasutada olemasoleva protsessi biomeetrilise isikutuvastuse sammu ja jätta operaatori külastamine ära. Rakenduse loomisel saaks seda müüa ka teisetele e-teenuste pakkujatele tootena ja teenida tulu iga biomeetrilise isikutuvastuse pealt.

Kuna loodud lahendus arendati samasse rakendusse, kuhu on arendatud ka täiskasvanute biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessid, siis leidis autor, et saaks luua juurde uue protsessi, mis lahendaks järgneva probleemi. Hetkel on täiskasvanutel biomeetrilise isikutuvastuse abil võimalik Smart-ID konto registreerida ainult siis, kui varasemalt on Smart-ID konto juba loodud. Tulevikulahendus oleks protsess, kus ei pea kasutajal varasemat kontot olema. Kuna Smart-ID konto registreerimiseks on kohtustulik nelja silma printsiip, siis tulevikulahenduse puhul kaoks olemasolev eelmise kontoga isikutuvastastamise samm ära, mis oli üheks kohustuslikuks silmapaariks. Selle asemele tuleks kasutada mingit muud viisi isikutuvastuseks, mille

üheks lahenduseks võiks olla näiteks eIDAS (*Electronic IDentification, Authentication and trust Services*) [23].

Hetkel on arendatud rakenduses andmebaasile juurdepääsuks kasutusel JDBC (*Java Database Connectivity*) programmeerimisliides, mis pakub meetodeid andmebaasis olevate andmete päringute tegemiseks ja uuendamiseks. Antud rakenduses tehakse palju päringuid andmebaasitabelite vastu, mis on omavahel seotud. Autori arvates võiks selle tulevikus välja vahetada mugavama JPA (*Java Persistence API*) programmeerimisliidese vastu, mille näide on toodud joonisel (vt Joonis 7). Peamine eelis arendajate jaoks JPA valiku puhul on koodi kirjutamine, kus saab rakendada objektorienteeritud põhimõtteid ja kasutada parimaid tavasid nii, et ei peaks muretsema andmebaasi süntaksi pärast. Tänu sellele saab soovitud rakendus või ülesanne arendatud kiiremini. Lisaks oma tugevale raamistikule aitab JPA kasutamine koodivigu vältida. JDBC kasutamine on hea oma lihtsuse poolest ja sobivam kasutada lihtsamate rakenduste puhul. JDBC programmeerimisliidese näide on toodud joonisel (vt Joonis 8). [24]

```
public interface SessionDao extends JpaRepository<ValidationSession, Integer>
{
    Optional<Session> findByUuid(UUID robSessionUUID);
}
```

Joonis 7. SessionDao klassi näide kasutades JPA programmeerimisliidest.

```
@Repository
public class SessionDao extends PostgresBaseDao {

    public SessionDao(DataSource dataSource) {
        super(dataSource);
    }

    public Session getSessionByUuid(UUID robSessionUUID) {
        try {
            return getJdbcTemplate().queryForObject("SELECT * FROM session
WHERE rob_session_uuid = ?",
                    BeanPropertyRowMapper.newInstance(Session.class),
                    robSessionUUID.toString(), action.name());
        } catch (EmptyResultDataAccessException ex) {
            throw new ServiceException(ServiceErrorCode.SESSION_MISSING,
"Session is missing.", ex);
        }
    }
}
```

Joonis 8. SessionDao klassi näide kasutades JDBC programmeerimisliidest.

6 Kokkuvõte

Bakalaureusetöö eesmärgiks oli luua biomeetrilise isikutuvastuse abil noorukite Smart-ID registreerimisprotsess. Olemasolevate Smart-ID registreerimisprotsesside peamisteks probleemideks on komplitseeritud registreerimisprotsess noorukitele ning pangakontoris registreerides lisaks ka liigsed kulud pangale.

Eesmärgi saavutamiseks uuriti olemasolevaid noorukitele loodud Smart-ID registreerimisprotsesse ja täiskasvanutele loodud biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessi. Seejärel tutvus autor olemasolevate nõuetega ja analüüsis, kuidas saaks olemasolevaid protsesse noorukite biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessi loomiseks kasutada. Analüüsi tulemusel selgus, et olemasolevad Smart-ID konto registreerimisprotsessid koosnevad sarnastest sammudest ja neid saab ära kasutada ka noorukite biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessi loomiseks.

Seejärel arendati nõuetele ja analüüsi tulemustele vastav kood. Kliendini viidi iga arendusetapi lõpus töötav kood ja testid, mida oli võimalik kasutusse võtta. Pärast minimaalse töötava koodi üleandmist tegeleti koodi refaktoreerimise, puhastamise ja veaolukordade parema lahendamisega. Tulemina valmis serveripoolne liides, mida SK ID Solutions AS kasutab noorukite biomeetrilise isikutuvastusega Smart-ID registreerimisprotsessi jaoks. Loodud lahendus aitab noorukitel registreerida Smart-ID konto lihtsamalt, hoides kokku raha ja aega. Tulemuste analüüsist selgus, et loodud protsessi äriline kasu väljendub peamiselt Smart-ID tehingute arvu kasvus ja protsessi biomeetrilise isikutuvastuse sammu saab ära kasutada ka mobiil-ID konto registreerimisprotsessis või teistes protsessides.

Bakalaureusetöö tulemused näitavad, et Smart-ID konto registreerimisprotsessid koosnevad sarnastest sammudest, mis on erinevate sisemiste protsesside ja järjekordadega. Tulevikus oleks võimalik teha põhjalikum analüüs, kus uuritakse kõikide sammude sisemisi protsesse täpsemalt. Analüüsi tulemustest oleks võimalik järeldada, kas ja kuidas saaks osad Smart-ID registreerimisprotsesside serveripoolsed sammud

ümber arendada selliselt, et päringute tegemiseks oleks võimalik kasutada ühiseid otspunkte. See aitaks vähendada otspunktide arvu ja korduvat koodi rakenduses.

Kasutatud kirjandus

[1] ID, „Tutvustus,“ [Võrgumaterjal]. Available: https://www.id.ee/rubriik/tutvustus-id-kaart/. [Kasutatud 22 04 2022].

[2] ID, „Smart-ID,“ [Võrgumaterjal]. Available: https://www.id.ee/artikkel/smart-id/. [Kasutatud 02 04 2022].

[3] Smart-ID, „Smart-ID: uue põlvkonna isikutuvastusteenus!,“ 10 04 2022. [Võrgumaterjal]. Available: https://www.smart-id.com/et/e-teenuste-pakkujale/. [Kasutatud 10 04 2022].

[4] SK ID Solutions AS, „Aasta 2017,“ 2017. [Võrgumaterjal]. Available: https://www.skidsolutions.eu/ettevottest/ajalugu/aasta-2017/. [Kasutatud 23 03 2022].

[5] Smart-ID, „Smart-ID - nutikas isikutuvastus,“ [Võrgumaterjal]. Available: https://www.smart-id.com/et/. [Kasutatud 05 03 2022].

[6] SK ID Solutions AS, „eID scheme: SMART-ID,“ 19 10 2021. [Võrgumaterjal]. Available: https://www.ria.ee/sites/default/files/content-editors/EID/smart-id-skeemi-kirjeldus-abiv.pdf. [Kasutatud 23 03 2022].

[7] Cybernetica AS, „SplitKey,“ [Võrgumaterjal]. Available: https://cyber.ee/products/splitkey. [Kasutatud 23 03 2022].

[8] M. Laanemägi, „Mis vahe on Mobiil-ID ja Smart-ID teenustel ning kas need asendavad „päris“ ID-kaarti?,“ 27 02 2018. [Võrgumaterjal]. Available: https://digitark.ee/mis-vahe-mobiil-id-ja-smart-id-teenustel-ning-kas-need-asendavad-paris-id-kaarti/. [Kasutatud 05 03 2022].

[9] I. Global, „Biometric security systems: a guide to devices, fingerprint scanners and facial recognition access control,“ 2016. [Võrgumaterjal]. Available: https://www.ifsecglobal.com/global/biometric-security-systems-guide-devices-fingerprint-scanners-facial-recognition/. [Kasutatud 2022 03 22].

[10] V. Zorkadis ja P. Donos, „On biometrics-based authentication and identification from a privacy-protection perspective: Deriving privacy-enhancing requirements,“ *Information Management & Computer Security,* pp. 125-137, 2004.

[11] A. Jain, L. Hong ja S. Pankanti, „Biometric identification,“ *Communications of the ACM,* kd. 98, p. 90, 2000.

[12] IProov, „iProov Face Verifier,“ [Võrgumaterjal]. Available: https://www.iproov.com/iproov-system/iproov-products-for-biometric-authentication/iproov-face-verifier. [Kasutatud 03 04 2022].

[13] DocuSign, „Sign documents and verify identity with the IDnow video service to issue qualified signatures,“ 07 04 2022. [Võrgumaterjal]. Available: https://support.docusign.com/s/document-item?language=en_US&bundleId=yca1573855023892&topicId=qtp1635247113600.html&_LANG=enus. [Kasutatud 10 04 2022].

[14] ID, „Dokobiti arendajasõbralikud autentimise ja digiallkirjastamise lahendused,“ [Võrgumaterjal]. Available: https://www.id.ee/artikkel/dokobiti-arendajasobralikud-autentimise-ja-digiallkirjastamise-lahendused/. [Kasutatud 10 04 2022].

[15] DocuSign, „The eSignature solution trusted by hundreds of millions of users,“ 2022. [Võrgumaterjal]. Available: https://www.docusign.com/products/electronic-signature. [Kasutatud 10 04 2022].

[16] D. Michalski, S. Y. Yiu ja C. Malec, „The Impact of Age and Threshold Variation on Facial Recognition Algorithm Performance Using Images of Children,“ *IEEE,* kd. 224, p. 217, 2018.

[17] J. Highsmith ja A. Cockburn, „Agile software development: the business of innovation,“ *Computer,* pp. 120-127, 2001.

[18] A. Vartan ja J.Brinkerhoff, „Minimum Viable Product: A maximally misunderstood idea,“ Slalom, 2022. [Võrgumaterjal]. Available: https://www.slalom.com/insights/mvp-maximally-misunderstood-term. [Kasutatud 29 03 2022].

[19] E. Allen, „InfoWorld,“ 01 01 2000. [Võrgumaterjal]. Available: https://www.infoworld.com/article/2076275/behold-the-power-of-parametric-polymorphism.html. [Kasutatud 21 03 2022].

[20] T. Hamiltion, „Unit Testing Tutorial: What is, Types, Tools & Test EXAMPLE,“ 16 04 2022. [Võrgumaterjal]. Available: https://www.guru99.com/unit-testing-guide.html. [Kasutatud 22 04 2022].

[21] T. Hamilton, „Integration Testing: What is, Types, Top Down & Bottom Up Example,“ 16 04 2022. [Võrgumaterjal]. Available: https://www.guru99.com/integration-testing.html. [Kasutatud 22 04 2022].

[22] SK ID Solutions AS, „Smart-ID teenuse hinnakiri,“ 01 01 2017. [Võrgumaterjal]. Available: https://www.skidsolutions.eu/teenused/hinnakiri/smart-id/. [Kasutatud 18 04 2022].

[23] RIA, „eIDAS,“ 09 06 2021. [Võrgumaterjal]. Available: https://www.ria.ee/et/riigi-infosusteem/usaldusteenused/eidas.html. [Kasutatud 20 03 0222].

[24] baeldung, „A Comparion Between JPA and JDBC,“ 5 12 2021. [Võrgumaterjal]. Available: https://www.baeldung.com/jpa-vs-jdbc. [Kasutatud 18 04 2022].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

Mina, Hugo Valk

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Biomeetrilise isikutuvastuse abil noorukite Smart-ID registreerimisprotsessi loomine", mille juhendaja on Tarvo Treier
   1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
   1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi

08.05.2022

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 loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – SessionService klass

```
@RequiredArgsConstructor
public class SessionService {

    private final SessionDao sessionDao;
    private final SessionPersonDao sessionPersonDao;

    public SessionWrapper getSessionWrapperByUuidAndActionForUpdate(UUID
robSessionUUID, ServiceAction serviceAction) {
        Session session =
sessionDao.getSessionByUuidAndActionForUpdate(robSessionUUID, serviceAction);
        List<SessionPerson> sessionPersons =
sessionPersonDao.getSessionPersonsBySessionId(session.getId());
        return toSessionWrapper(session, sessionPersons);
    }
}
```

Lisa 3 – AbstractSessionService klass

```
@RequiredArgsConstructor
public abstract class AbstractSessionService<T extends SessionWrapper> {

    protected final SessionDao sessionDao;
    protected final SessionPersonDao sessionPersonDao;

    public abstract T getSessionWrapperByUuidAndActionForUpdate(UUID
robSessionUUID, ServiceAction serviceAction);
}
```

Lisa 4 – AdultSessionService klass

```
@Service
public class AdultSessionService extends
AbstractSessionService<SessionWrapper> {

    public AdultSessionService(SessionDao sessionDao, SessionPersonDao
sessionPersonDao) {
        super(sessionDao, sessionPersonDao);
    }

    @Override
    public SessionWrapper getSessionWrapperByUuidAndActionForUpdate(UUID
robSessionUUID, ServiceAction serviceAction) {
        Session session =
sessionDao.getSessionByUuidAndActionForUpdate(robSessionUUID, serviceAction);
        List<SessionPerson> sessionPersons =
sessionPersonDao.getSessionPersonsBySessionId(session.getId());
        return toSessionWrapper(session, sessionPersons);
    }
}
```

Lisa 5 – MinorSessionService klass

```
@Service
public class MinorSessionService extends
AbstractSessionService<MinorSessionWrapper> {

    private final SessionPersonLegalGuardianDao
sessionPersonLegalGuardianDao;

    public MinorSessionService(SessionDao sessionDao, SessionPersonDao
sessionPersonDao, SessionPersonLegalGuardianDao
sessionPersonLegalGuardianDao) {
        super(sessionDao, sessionPersonDao);
        this.sessionPersonLegalGuardianDao = sessionPersonLegalGuardianDao;
    }

    @Override
    public MinorSessionWrapper getSessionWrapperByUuidAndActionForUpdate(UUID
robSessionUUID, ServiceAction serviceAction) {
        Session session =
sessionDao.getSessionByUuidAndActionForUpdate(robSessionUUID, serviceAction);
        List<SessionPerson> sessionPersons =
sessionPersonDao.getSessionPersonsBySessionId(session.getId());
        List<SessionPersonLegalGuardian> sessionPersonLegalGuardians =
sessionPersonLegalGuardianDao.getSessionPersonLegalGuardiansBySessionId(sessi
on.getId());
        return toMinorSessionWrapper(session, sessionPersons,
sessionPersonLegalGuardians);
    }
}
```