

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Rando Tiri 223190IABB
Stella Marie Tohver 233672IABB

**Ravimite triipkoodipõhise tuvastamise ja
raviskeemide haldamise rakenduse analüüs ja
prototüübi arendamine**

Bakalaureusetöö

Juhendajad: Gunnar Piho

PhD

Viljam Puusep

MSc

Tallinn 2026

Autorideklaratsioon

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

Autorid: Rando Tiri, Stella Marie Tohver

17.05.2026

Lühikokkuvõte

Tavakasutajal puudub lihtne ja kiire viis tuvastada ravimit selle pakendil oleva triipkoodi alusel ning saada koheselt kätte usaldusväärne põhiinfo. Kuigi Eesti riiklik ravimiregister sisaldab vajalikke andmeid, ei ole see optimeeritud igapäevaseks mobiilseks kasutuseks. Ravimi leidmine eeldab käsitsi otsimist ning info esitusviis ei toeta kiiret kasutust. Käesolev bakalaureusetöö analüüsib seda probleemi ning kirjeldab prototüüpse rakenduse kavandamist ja arendamist Design Science lähenemise järgi.

Töö tulemuseks on MedScan nimeline prototüüpne mobiilirakendus, mis võimaldab ravimite tuvastamist EAN-13 ja DataMatrix triipkoodide skaneerimise või nimepõhise otsingu abil. Rakendus sisaldab koduapteegi halduse funktsionaalsust, mis võimaldab jälgida ravimite koguseid, aegumiskuupäevi ning hallata raviskeeme koos ajastatud meeldetuletustega. Lisaks näeb kasutaja ravimi ametlikku infolehte. Süsteem toetab ka mitme kasutajaprofiili haldust ja võimaldab hallata näiteks pereliikmete ravimeid.

Töö järelduste kohaselt on triipkoodipõhine ravimite tuvastamine tehniliselt teostatav ning pakub selget lisaväärtust tavakasutajale. See parandab ravimiinfo kättesaadavust ja raviskeemi järgimise mugavust.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 42 leheküljel, 6 peatükki, 16 joonist, 4 tabelit.

Abstract

Analysis and Prototype Development of a Barcode-based Medication Identification and Medication Management Application

Ordinary users lack a simple and quick way to identify a medication based on the barcode on its packaging and immediately access reliable basic information. Although Estonia's national drug registry contains the necessary data, it is not optimized for everyday mobile use. Finding a medication requires manual searching, and the way the information is presented does not support quick access. This bachelor's thesis analyzes this problem and describes the design and development of a prototype application using the Design Science approach. This involves systematically mapping existing processes, designing and implementing a solution, and evaluating it iteratively through functional testing and use-case-based validation.

The result of this thesis is a prototype mobile application called MedScan, which enables the identification of medications by scanning EAN-13 and DataMatrix barcodes or through a name-based search. The application includes home medicine cabinet management functionality, which allows users to track medication quantities and expiration dates and manage treatment regimens with scheduled reminders. Each medication card displays the active substance and strength, and links directly to the official product information sheet on the Estonian State Agency of Medicines website. The system also supports the management of multiple user profiles, allowing users to manage, for example, medications for family members. When a medication is running low, the application notifies the user, so they can consider purchasing a new package.

According to the study's findings, barcode-based medication identification is technically feasible and offers clear added value to the average user by improving access to medication information and the convenience of adhering to treatment regimens.

The thesis is in Estonian and contains 42 pages of text, 6 chapters, 16 figures, 4 tables.

Lühendite ja mõistete sõnastik

API	Application Programming Interface
ATC	Anatomical Therapeutic Chemical
BPMN	Business Process Model and Notation
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DataMatrix	Kahemõõtmeline maatrikskood
DTO	Data Transfer Object
EAN-13 numbriline kood	European Article Number 13-kohaline numbriline kood
FMD	Falsified Medicines Directive 2011/62/EU, Euroopa Liidu võltsimisevastase direktiiv
GS1	Global Standard One
GTIN-13	Global Trade Item Number 13-kohaline numbriline kood
HTTP	HyperText Transfer Protocol
JWT	JSON Web Token
JSON	JavaScript Object Notation
MAUI	Multi-platform App User Interface
MVP	Minimum Viable Product
PDF	Portable Document Format
REST	Representational State Transfer
SMTP	Simple Mail Transfer Protocol
UI	User Interface
UML	Unified Modeling Language
ZXing	Zebra Crossing

Sisukord

Jooniste loetelu	8
Tabelite loetelu	9
1 Sissejuhatus	10
1.1 Üldine taust.....	10
1.2 Probleem ja projekti eesmärk	10
1.3 Töö edasine struktuur	11
2 Taust	12
2.1 Triipkoodid ja nende kasutamine ravimitööstuses	12
2.2 Eesti ravimiregister ja ravimiandmete kättesaadavus.....	13
2.3 Olemasolevad lahendused ja turuolukord.....	13
2.4 Äriline vajadus ja sihtrühm	14
3 Metoodika.....	16
3.1 Uurimismeetodid	16
3.2 Andmete kogumine ja kasutajavajaduste kaardistamine	17
3.3 Arendusvahendid ja tehnoloogiad	17
3.4 Arendusprotsess.....	18
4 Tulemused	19
4.1 Algusprotsess.....	19
4.1.1 Persoonad ja kasutajavood.....	19
4.1.2 Kasutajalood	20
4.1.3 Projekti seadistamine	23
4.2 Rakenduse loomine	23
4.2.1 Figma ja Lovable prototüüpide loomine ja testimine.....	23

4.2.2	Prototüübi loomine	26
4.2.3	Rakenduse arhitektuur	27
4.2.4	Andmemudel	29
4.2.5	Prototüübi testimine.....	30
4.3	Realiseeritud funktsionaalsused	33
4.3.1	Autentimine	33
4.3.2	Triipkoodi skaneerimine ja ravimi tuvastamine	34
4.3.3	Ravimi infoleht.....	35
4.3.4	Raviskeem ja meeldetuletused.....	36
4.3.5	Madala koguse teavitused.....	37
5	Analüüs ja järeldused	38
5.1	Tehnilise teostuse põhjendus	38
5.2	Hinnang projekti teostamise protsessi kohta	39
5.3	Projekti kitsaskohad ja õnnestumised.....	39
5.4	Lahenduse äriline ja praktiline põhjendatus	40
5.5	Projekti tulevikuplaanid.....	41
5.6	Autorite eneseanalüüs.....	43
5.6.1	Stella eneseanalüüs	43
5.6.2	Rando eneseanalüüs.....	44
6	Kokkuvõte	46
	Kasutatud kirjandus	47
	Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	49
	Lisa 2 – Andmebaasis olevate ravimite karbid ja triipkoodid	50
	Lisa 3 – Kuvatõmmised valminud prototüübist	52

Jooniste loetelu

Joonis 1. MedScan kasutusjuhu diagramm.....	22
Joonis 2. Figma Make-is koostatud disain.	24
Joonis 3. Lovable-is koostatud disain.....	25
Joonis 4. MedScan süsteemi arhitektuuriskeem.	28
Joonis 5. Klassidiagramm MedScan-i peamiste andmebaasimudelite kohta.	30
Joonis 6. Autentimise jadadiagramm	34
Joonis 7. Skaneerimise jadadiagramm.	35
Joonis 8. Ravimi infolehe BPMN-diagramm.	36
Joonis 9. Ravimi lisamise BPMN-diagramm.	36
Joonis 10. Andmebaasis olevad EAN-13 triipkoodiga ravimid.	50
Joonis 11. Andmebaasis olevad DataMatrix triipkoodiga ravimid.	51
Joonis 12. Kuvatõmmised prototüübi autentimise lehtedest.	53
Joonis 13. Kuvatõmmised prototüübi skaneerimise lehtedest.....	54
Joonis 14. Kuvatõmmised prototüübi ravimi infolehest.....	55
Joonis 15. Kuvatõmmised prototüübi lisa nimekirja ja ravimid lehtedest.	56
Joonis 16. Kuvatõmmised prototüübi meeldetuletustest.	57
Joonis 17. Kuvatõmmis prototüübi madala koguse teavitusest.	58
Joonis 18. Kuvatõmmised prototüübi hoolealuste haldamise lehest.	59

Tabelite loetelu

Tabel 1. Olemasolevate ravimihaldusrakenduste võrdlus MedScani kavandatavate funktsionaalsustega.....	14
Tabel 2. Design Science lähenemise rakendamine käesolevas töös.....	16
Tabel 3. MedScan kasutajapersoonad.	19
Tabel 4. MedScan kasutajavood ja realiseerimise staatus.....	20
Tabel 5. Funktsionaalsed nõuded.	20
Tabel 6. Mittefunktsionaalsed nõuded.	21
Tabel 7. MedScan rakenduse arhitektuur.	27
Tabel 8. Järeldatavate tulemused.....	31

1 Sissejuhatus

1.1 Üldine taust

Ravimite kasutamine on igapäevane reaalsus miljonite inimeste jaoks üle maailma. Ravimiinfo õigeaegne ja lihtne kättesaadavus on oluline nii ravi ohutuse kui ka tulemuslikkuse seisukohalt. Eestis on ravimite ametlik teave koos pakendi infolehe ja omaduste kokkuvõttega kättesaadav riikliku ravimiregistri veebilehel. Ravimiregister on Ravimiameti hallatav riiklik andmekogu, mis sisaldab põhjalikku ja usaldusväärset infot kõigi Eestis müügiloa saanud ravimite kohta (Ravimiamet, i.a.-a).

1.2 Probleem ja projekti eesmärk

Tavakasutajal puudub lihtne ja kiire viis tuvastada konkreetset ravimit selle pakendil oleva triipkoodi alusel ning saada kohe kätte ravimi põhiinfo ja viide ametlikule infolehele. Ravimiinfo on küll olemas riiklikus andmeallikas, kuid selle leidmine eeldab ravimi nime käsitsi otsimist ning info on esitatud viisil, mis ei ole optimeeritud kiireks igapäevaseks mobiilseks kasutamiseks. Selle tulemusena on ravimi tuvastamine ja esmase informatsiooni leidmine kasutaja jaoks aeganõudev ja ebamugav, mis võib mõjutada ravimite õiget ja ohutut tarvitamist. Mida kiiremini ja lihtsamalt jõuab kasutaja usaldusväärse ravimiinfoni, seda tõenäolisem on ravimi korrektne kasutamine ning aegunud või sobimatu ravimi varajane tuvastamine.

Töö eesmärk on analüüsida triipkoodipõhise ravimituvastuse vajadust ning kavandada ja arendada sobiv tarkvaraline lahendus. Rakendus võimaldab kasutajal tuvastada ravim pakendilt triipkoodi skaneerimise kaudu ning kuvada selle kohta informatsioon koos viitega ametlikule infolehele. Lisaks on eesmärgiks kavandada ja realiseerida lisafunktsionaalsused, mis võimaldavad kasutajal siduda tuvastatud ravimeid oma ravimite nimekirjaga ning lisada nende juurde märkmeid ja manustamisega seotud informatsiooni.

Töö artefaktiks on MedScan nimeline prototüüpne mobiilirakendus, mis ühendab triipkoodipõhise ravimituvastuse ja ravimihalduse ühtseks tervikuks. Rakendus on arendatud .NET MAUI Blazor Hybrid raamistikus ning demonstreerib, et selline lahendus on tehniliselt teostatav.

Töö äriline väärtus seisneb selles, et automatiseeritud ravimiinfo kuvamine vähendab inimlikke vigu, säästab kasutajate aega ning toetab ravimite korrektset ja ohutut tarvitamist.

1.3 Töö edasine struktuur

Lisaks sissejuhatusle on töös veel viis peatükki. Tausta peatükis antakse ülevaade triipkoodidest, Eesti ravimiregistrist, olemasolevatest lahendustest ning ärilisest vajadusest. Metoodika osas kirjeldatakse detailselt uuritavat objekti, tuuakse välja kõik kasutatud tööriistad ning tööprotsess. Tulemuste peatükis kirjeldatakse arenduse kulgu algusest lõpuni. Persoonade loomist, kasutajavoogude ja kasutajalugude defineerimist ning Figma koostatud tüpograafiat ja kujundust, lõpuks rakenduse prototüübi realiseerimist. Analüüsi ja järelduste peatükk sisaldab tehnilisi põhjendusi ja hinnangut. Hinnatakse projekti teostamise protsessi ning tulevikuplaane. Samuti leiab sellest peatükist autorite eneseanalüüsi.

2 Taust

2.1 Triipkoodid ja nende kasutamine ravimitööstuses

Triipkood on kodeerimissüsteem, mis identifitseerib kodeeritud ühemõõtmeliste joonte ehk 1D triipkoodide või kahemõõtmeliste mustrite ehk 2D triipkoodide kujul toote andmeid (GS1 Eesti, 2015, lk 7). Triipkoodide rahvusvaheliste standardite haldamisega tegeleb ülemaailmne mittetulunduslik organisatsioon GS1, mis on loonud ühtse globaalse numbrisüsteemi toodete üheseks identifitseerimiseks (GS1 Eesti, 2015, lk 4).

EAN-13 on 13-kohaline 1D triipkood, mida kasutatakse ravimipakenditel peamiselt toote identifitseerimiseks jaemüügis, apteekide laohalduses ning logistikas. See sisaldab unikaalset GTIN-13 koodi, mis võimaldab pakendit rahvusvaheliselt tuvastada. Kood koosneb neljast osast: GS1 riigiprefiksist, mis tähistab riiki, kus toode on registreeritud, ettevõttekoodist, tootekoodist ning kontrolljärgust (GS1 Eesti, 2015, lk 11-12). EAN-13 esineb ravimitööstuses peamiselt käsimüügiravimite pakenditel. EAN-13 suur eelis on lai levik ja lihtne loetavus igasuguse skänneriga. Koodi peamine piirang on väike andmemaht. EAN-13 sisaldab ainult tootetunnust ning ei mahuta lisainfot nagu partiinumber või aegumiskuupäev (EFPIA, 2008).

DataMatrix on 2D maatrikskood, mis on loodud peamiselt tervishoiusektori vajadusi silmas pidades (GS1 Eesti, 2015, lk 8). Selle kasutamine on olnud alates 9. veebruarist 2019. aastast Euroopa Liidus kohustuslik retseptiravimite pakenditel Euroopa Liidu võltsimisevastase direktiivi FMD alusel (Pharmaceutical Society of Ireland, i.a.). Erinevalt EAN-13-st kodeerib DataMatrix andmeid elementidesse, mitte triipudesse, mahutades seeläbi oluliselt rohkem informatsiooni, mis on ravimite ohutuse ja jälgitavuse seisukohalt kriitilise tähtsusega (GS1 Eesti, 2015, lk 11-12). Direktiivi nõuete kohaselt peab DataMatrix kood sisaldama nelja kohustuslikku andmeelementi: tootekood (GTIN), seerianumber, partiinumber ning aegumiskuupäev (Pharmaceutical Society of Ireland, i.a.).

2.2 Eesti ravimiregister ja ravimiandmete kättesaadavus

Eesti ravimiregister on Ravimiameti hallatav riiklik andmekogu, mis on loodud ravimiseaduse § 79 lõike 1 alusel ning kuulub riigi infosüsteemi koosseisu (Vabariigi Valitsus, 2016). Registri eesmärk on pidada arvestust kõigi Eestis müügiluba omavate ja turustatavate ravimite üle, identifitseerida neid tervishoius kasutatavates infosüsteemides ning tagada avalikkusele asjakohase teabe kättesaadavus (Vabariigi Valitsus, 2016).

Ravimiregister sisaldab kõiki Ravimiameti või Euroopa Komisjoni väljastatud müügiloaga ravimeid (Ravimiamet, 2020). Ravimeid on registrist võimalik otsida ravimi nimetuse, toimeaine eesti-, inglise- või ladinakeelse nimetuse ning ATC koodi alusel (Apotheka, i.a.). Andmekogust on võimalik leida ravimi toimeainet, ravimivormi, tugevust ja muud vajalikku infot (Ravimiamet, i.a.).

Kuigi ravimiregister sisaldab põhjalikku ja usaldusväärset infot, puudub sellel võimalus teostada triipkoodipõhist päringut. Seetõttu on praegu toetamata igapäevane mobiilne lahendus, kus kasutaja saab kiiresti ravimi kohta infot leida pakendil oleva triipkoodi alusel.

2.3 Olemasolevad lahendused ja turuolukord

Rahvusvaheliselt on saadaval mitmeid ravimihalduslahendusi, näiteks Medisafe Medication Reminder (Medisafe, 2026a; Medisafe, 2026b), MyTherapy Medication Tracker (SmartPatient GmbH, 2026a; SmartPatient GmbH, 2026b), Capsule (Capsule, 2026) ja Whobert (Whobert, 2026), kuid need ei ole kohandatud Eesti ravimiregistri andmetele ega arvesta kohalikku regulatiivset konteksti. Ükski vaadeldud lahendustest ei toeta EAN-13 ega DataMatrix triipkoodide skaneerimist. Samuti puudub ravimi ametliku infolehe link. Tabelis 1 on esitatud nimetatud rakenduste võrdlus MedScani kavandatavate funktsionaalsustega.

Tabel 1. Olemasolevate ravimihaldusrakenduste võrdlus MedScani kavandatud funktsionaalsustega.

Funktsioon	Medisafe Medication Reminder	MyTherapy Medication Tracker	Capsule	Whobert	MedScan
Triipkoodi skännimine (EAN-13 + Datamatrix tugi)	Ei	Ei	Ei	Ei	Jah
Eesti ravimiregister	Ei	Ei	Ei	Ei	Jah
Ravimite andmebaas	Jah	Ei	Ei	Ei	Jah
Link ametlikule infolehele	Ei	Ei	Ei	Ei	Jah
Ravimi kirjeldus	Jah	Ei	Ei	Ei	Jah
Meeldetuletused	Jah	Jah	Jah	Jah	Jah
Koduapteegi haldus	Jah	Jah	Jah	Jah	Jah
Koguste jälgimine	Jah	Jah	Ei	Jah	Jah
Aegumiskuu-päeva jälgimine	Ei	Ei	Ei	Ei	Jah
Märkmete lisamine	Jah	Ei	Jah	Jah	Jah
Hoolealuste haldus	Ei	Ei	Ei	Jah	Jah
Manustamise logi	Jah	Jah	Jah	Jah	Jah
Rakenduse hind	Tasuta	Tasuta	Osaliselt tasuta, piirangutega	7.99 €/month või 69.99 €/year	Tasuta
Platvorm	iOS / Android	iOS / Android	iOS	iOS	iOS / Android

2.4 Äriline vajadus ja sihtrühm

Ravimite vale või ebaregulaarne tarbimine on ülemaailmselt tänaseni probleem. Ravimite mittetarbimine põhjustab Euroopas hinnanguliselt 200 000 surma ja 125 miljardit eurot kulusid aastas (ENABLE COST Action, 2024). Uuringud näitavad, et ainult pooled patsiendid tarbivad ravimeid vastavalt ettekirjutusele (ENABLE COST Action, 2024). Eestis on ravimite kättesaadavus hea, kuid igapäevane ravimihaldus, aegumiskuupäevade jälgimine, raviskeemide pidamine ja ravimiinfo leidmine toetub suures osas kasutaja enda initsiatiivile. Lihtsa ja usaldusväärse digitaalse abivahendi puudumine loob selge vajaduse MedScani sarnase lahenduse järele.

MedScani sihtrühm on nutitelefone kasutavad isikud, kes tarbivad regulaarselt ravimeid, soovivad oma koduapteegi üle paremat ülevaadet pidada või hallata hoolealuste ravimiskeemi.

Käesolev töö realiseerib tasuta kasutatava prototüübi. Tulevikus on võimalik kaaluda lisafunktsionaalsuste tasustamist, integratsiooni apteegisüsteemidega või rakenduse pakkumist tervishoiuasutustele.

3 Metoodika

3.1 Uurimismeetodid

Töö metoodiliseks aluseks on Design Science lähenemine, mille kohaselt omandatakse teadmised probleemi kohta rakenduse kavandamise ja arendamise käigus (Hevner et al., 2004). Design Science eristab end klassikalisest tarkvara arendusest selle poolest, et uurimisprotsess ei piirdu üksnes rakenduse valmistamisega, vaid hõlmab ka süstemaatilist valideerimist. Lahenduse sobivust hinnatakse kasutajate tegelike vajaduste suhtes nii eel- kui järeltestimise kaudu (Hevner et al., 2004).

Design Science lähenemine koosneb kuuest järjestikusest etapist. Probleemi tuvastamise etapis kaardistatakse uuritav valdkond ja sõnastatakse lahendamist vajav probleem. Eesmärkide defineerimise etapis tuletatakse probleemist konkreetset nõudeid loodavale artefaktile. Disaini ja arenduse etapis luuakse artefakt vastavalt püstitatud nõuetele. Demonstratsiooni etapis rakendatakse valminud artefakti reaalses kasutuskontekstis, et näidata selle toimivust probleemi lahendamisel. Hindamise etapis valideeritakse artefakti vastavust algselt püstitatud eesmärkidele ning kommunikatsiooni etapis vormistatakse tulemused teaduslikuks väljundiks. Tabelis 2 on näidatud, kuidas töö tegevused jagunevad nende etappide vahel ning millistes peatükkides iga etapp on kirjeldatud.

Tabel 2. Design Science lähenemise rakendamine käesolevas töös.

Design Science etapp	Tegevus	Peatükk
Probleemi tuvastamine	Ravimiinfo kättesaadavuse analüüs, turuolukord.	1.2, 2
Eesmärkide defineerimine	Persoonad, kasutajavood, kasutajalood.	4.1.1, 4.1.2
Disain ja arendus	Figma ja Lovable prototüübid, MAUI rakendus.	4.2
Demonstratsioon	Realiseeritud funktsionaalsused.	4.3
Hindamine	Eel- ja järeltestimine kasutajatega.	4.2.1, 4.2.5
Kommunikatsioon	Bakalauresetöö.	Kogu töö

3.2 Andmete kogumine ja kasutajavajaduste kaardistamine

Projekti algfaasis loodi esmalt persoonad, mis kirjeldasid rakenduse potentsiaalseid sihtkasutajaid. Persoonade põhjal koostati kasutajalood, mis aitasid määratleda, milliseid funktsionaalsusi peab rakendus täitma ning millised on kasutaja ootused ja vajadused. Kasutajalood struktureeriti GitLabi *issue*-dena, mis võimaldasid arenduse käiku süstemaatiliselt jälgida ja hallata. Persoonade ning kasutajalugude põhjal koostati detailne prompt, mille alusel genereeriti interaktiivsed visuaalsed prototüübid. Persoonad, kasutajavood ja prototüübid on lähemalt kirjeldatud peatükkides 4.1 ja 4.2.

Valminud prototüüpide põhjal viidi läbi eeltestimine, kus kasutajad said prototüüpidega tutvuda ning anda tagasisidet. Eeltestimise eesmärk oli hinnata, kas triipkoodipõhisel ravimituvastuse rakendusel on reaalne kasutajapõhi ning kas selline lahendus pakub kasutajatele tegelikku lisaväärtust. Eeltestimise positiivse tagasiside põhjal kinnitati, et rakendusel on praktiline väärtus, ning alustati päris rakenduse arendamisega.

Pärast prototüübi valmimist viidi läbi järeltestimine, kus kasutajad täitsid etteantud kasutusstsenaariumid ning andsid tagasisidet rakenduse kasutatavuse ja funktsionaalsuse kohta. Kogutud tagasiside põhjal hinnati, mil määral vastab realiseeritud lahendus alguses püstitatud kasutajavajadustele ning tehti vajalikud täiendused ja parandused.

3.3 Arendusvahendid ja tehnoloogiad

Rakenduse põhiraamistikuks valiti .NET MAUI Blazor Hybrid. See võimaldab luua C# keele ja Razor komponentide abil ühe koodibaasi põhjal nii mobiilirakenduse kui ka veebirakenduse. Raamistik valiti koostöös juhendajate Dr Gunnar Piho ja Viljam Puusepaga, kelle hinnangul oli see antud ülesande jaoks kõige sobivam valik. Lisaks toetas valikut töö autorite varasem kokkupuude .NET keskkonnaga. Tallinna Tehnikaülikooli õppekava aines ISA II (Tallinna Tehnikaülikool, 2023) omandati .NET raamistiku põhiteadmised ning Microsofti ametlikust dokumentatsioonist ja õpetustest saadi arenduse alustamiseks piisavad teadmised (Microsoft, 2025).

Rakenduse arendamisel kasutati järgmisi tehnoloogiaid ja tööriistu:

- ASP.NET Core – API loogika ja serveripoolne liidesekiht.
- ASP.NET Identity ja JWT – kasutajate autentimine ja autoriseerimine.

- Bizagi Modeler – töövooskeemide loomine BPMN formaadis.
- Docker – andmebaasi konteineriseerimine ja ühtse keskkonna tagamine.
- Enterprise Architect – UML diagrammide loomine.
- Figma – rakenduse tüpograafia loomine.
- Figma Make ja Lovable – tehisintellektipõhised disainitööriistad, millega loodi algsed interaktiivsed prototüübid.
- GitHub Copilot ja Codex – tehisintellektiassistendid koodiarenduses.
- GitLab – versioonihaldus ja meeskonnatöö korraldamine.
- .NET MAUI Blazor Hybrid – multiplatvormiline kasutajaliides iOS-i ja Androidi jaoks.
- PostgreSQL – relatsiooniline andmebaas ravimite ja kasutajaandmete hoidmiseks.
- Visual Studio – arenduskeskkond.
- ZXing – EAN-13 ja DataMatrix triipkoodide lugemine kaamera kaudu.

3.4 Arendusprotsess

Arendusfaasis arendati rakendus moodulite kaupa järgmises järjekorras: autentimine ja kasutajahaldus, ravimite andmemudel ja andmebaas, triipkoodide skaneerimise funktsionaalsus ning raviskeemi haldus koos meeldetuletustega. Arendus toimus iteratiivselt kogu projekti vältel. Arendustöös kasutati programmeerimist toetavaid generatiivse tehisintellekti tööriistu OpenAI Codex ja GitHub Copilot (OpenAI, 2026; GitHub, 2026). Nende abil loodi osa rakenduse programmikoodist ning funktsioonide esmased lahendused. Autorid kirjeldasid soovitud funktsionaalsust, hindasid tööriistade pakutud lahendusi, kohandasid loodud koodi rakenduse vajadustele vastavaks ning testisid selle toimimist.

Arendusprotsessi üksikasjalikum kirjeldus, sealhulgas meeskonna koostöömudel, tööruutin ja kogetud väljakutsed on esitatud peatükis 5.2.

4 Tulemused

4.1 Algusprotsess

4.1.1 Persoonad ja kasutajavood

Kasutajate vajaduste paremaks mõistmiseks loodi autorite poolt neli persoonat, mis kujutavad rakenduse tüüpilisi sihtkasutajaid. Persoonad aitavad arendusprotsessis hoida fookust tegelikel kasutajatel ning tagada, et tehtavad otsused lähtuvad kasutajate vajadustest, mitte üksnes tehnilistest võimalustest. Tabelis 3 on esitatud neli peamist persoonat koos nende eesmärkide, valupunktide ja prototüübis kaetud funktsionaalsustega.

Tabel 3. MedScan kasutajapersoonad.

Persoon	Eesmärgid	Valupunktid	Kaetud funktsionaalsus
Maie Tamm, 68, tavapatsient	Tuvastada ravim kiiresti; näha põhiinfot; pidada raviskeemi; saada meeldetuletusi	Ravimiregister on aeglane; infolehed keerulised; ununeb, kas ravim võeti	Triipkoodi skaneerimine; nime järgi otsing; ravimi põhiinfo ja link infolehele; raviskeem; manustamise logimine; meeldetuletused
Liis Tamm, 34, lapsevanem	Hallata enda ja lapse ravimeid eraldi; saada teavitusi lapse ravimi andmiseks	Infolehed raskesti leitavad; keeruline meele pidada, kas lapsele anti ravim	Profiilipõhine haldus; ravimi lisamine valitud profiilile; raviskeemi ajad ja meeldetuletused; manustamise ajalugu
Margus Järv, 29, lühiajaline ravikuur	Seadistada ajutine raviskeem kiiresti; saada lihtne ülevaade ravimist	Ravimiinfo leidmine tüütu; puudub keskne koht lühikuuri haldamiseks	Kiire skaneerimisvoog; ravimi lisamine sageduse ja kellaegadega; märkuste lisamine; „Minu ravimid“ vaade
Aivar Sepp, 45, hooldaja	Kontrollida ravimi õigsust enne manustamist; hallata mitut patsienti	Kätsi kontroll aeglane ja veaohlik; mitme patsiendi haldus keeruline	Osaliselt kaetud: profiilide ja skaneerimine tehniline alus on olemas

Tabelis 4 näeb personade põhjal koostatud kasutajavoogusid, mis kirjeldavad, kuidas iga persona rakenduses liigub. Kasutajavoogude koostamine aitas tuvastada olulised

vaated ja nende omavahelised seosed, et rakenduse navigatsioon on loogiline ja kasutajasõbralik.

Tabel 4. MedScan kasutajavood ja realiseerimise staatus.

Kasutajavoog	Kirjeldus	Staatus
Voog 1: Ravimi tuvastamine	Sisselogimine → skaneerimine või otsing → põhiinfo kuvamine → lisamine raviskeemi	Realiseeritud
Voog 2: Hoolealuse ravimi haldus	Profiili valik → ravimi lisamine → manustamisajad ja meeldetuletus → kinnitamine	Realiseeritud
Voog 3: Manustamise kinnitamine	Teavitus → annuse kinnitamine → logi salvestamine → ajaloo kuvamine	Realiseeritud
Voog 4: Parooli taastamine	„Unustasin parooli” → kood e-postile → koodi kontroll → uus parool	Realiseeritud
Voog 5: Koduapteegi haldus	Ravimi lisamine → koguse ja aegumise haldus → kirje uuendamine või kustutamine	Realiseeritud
Voog 6: Hooldaja töövoog (asutus)	Töötajakonto → patsientide koondnimekiri → skaneerimisel vastavuskontroll	Osaliselt realiseeritud / planeeritud
Voog 7: Veebikasutus	Rakenduse täisfunktsionaalsus veebikliendis	Planeeritud

4.1.2 Kasutajalood

Kasutajavoogude põhjal koostati kasutajalood, mis kirjeldavad konkreetseid funktsionaalsusi kasutaja vaatepunktist. Kasutajalugude põhjal määratleti süsteemi nõuded, mis jaotati funktsionaalseteks ja mittefunktsionaalseteks nõueteks. Funktsionaalsed nõuded kirjeldavad, mida süsteem peab tegema, mittefunktsionaalsed nõuded aga seda, kuidas süsteem peab toimima. Nõuded on esitatud tabelites 5 ja 6.

Tabel 5. Funktsionaalsed nõuded.

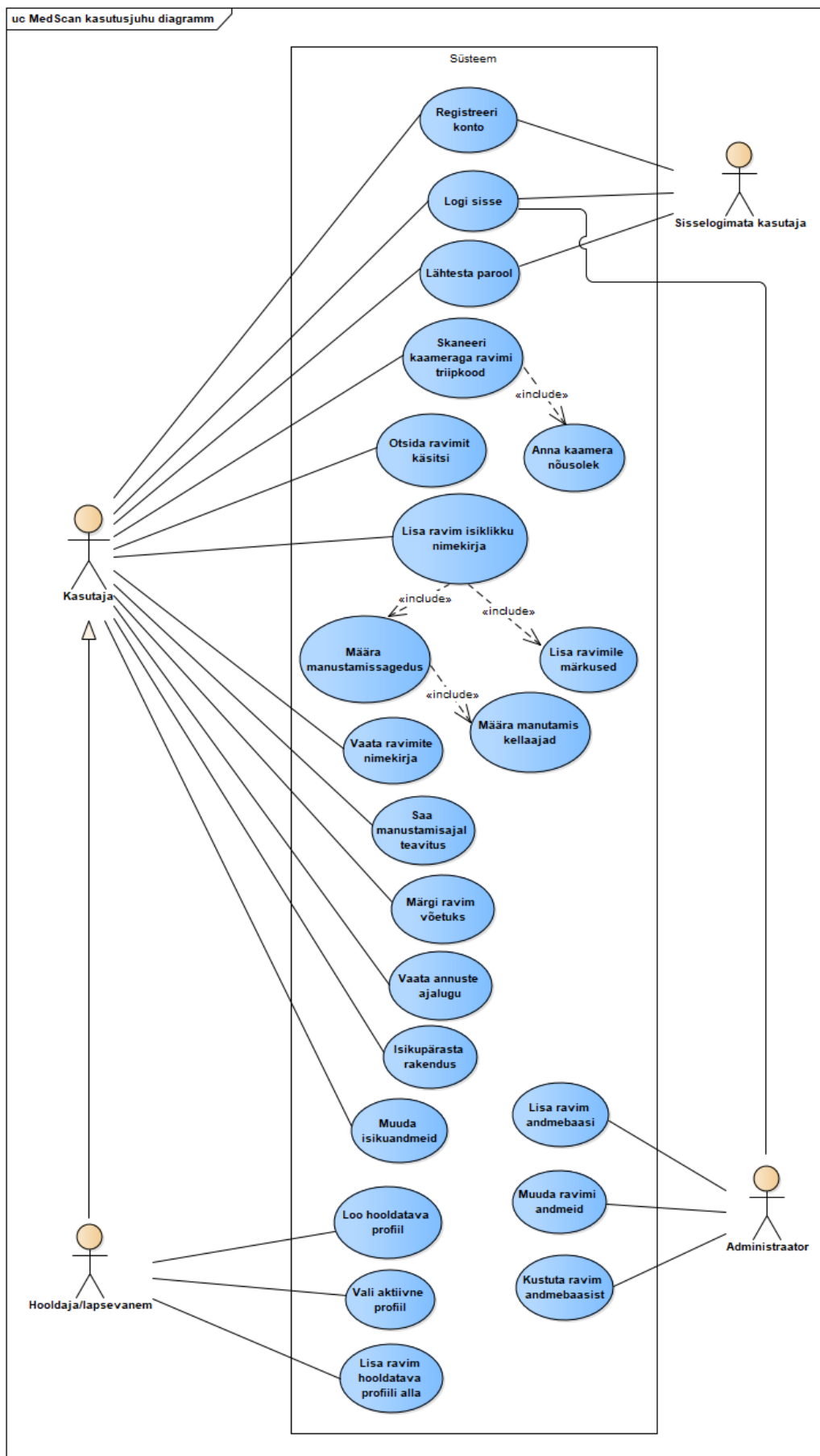
ID	Nõue
FN-01	Süsteem võimaldab kasutaja registreerimist e-posti ja parooliga.
FN-02	Süsteem võimaldab sisselogimist e-posti ja parooliga.
FN-03	Süsteem võimaldab parooli taastamist e-posti kaudu.
FN-04	Süsteem skaneerib EAN-13 triipkoodi kaameraga.
FN-05	Süsteem skaneerib DataMatrix triipkoodi kaameraga.
FN-06	Süsteem küsib kaameraõigust enne skaneerimist.
FN-07	Süsteem pakub käsitsi nime järgi otsimist, kui triipkoodi ei tuvastata
FN-08	Süsteem kuvab tuvastatud ravimi põhiinfot.
FN-09	Süsteem võimaldab kasutajal ravimiregistri ametiliku infolehe lugemist.
FN-10	Kasutaja saab lisada ravimi isiklikku nimekirja.
FN-11	Kasutaja saab määrata manustamissageduse ja kellaajad.
FN-12	Kasutaja saab lisada ravimile vabatekstilisi märkusi.
FN-13	Süsteem saadab kasutajale meeldetuletusi ravimi võtmiseks.
FN-14	Kasutaja saab kinnitada manustamise.
FN-15	Süsteem peab manustamise logi.
FN-16	Kasutaja saab luua hoolealuse profiile.

ID	Nõue
FN-17	Kasutaja saab lisada ravimi hoolealuse profiili alla.
FN-18	Süsteem saadab profiilipõhiseid meeldetuletusi.
FN-19	Süsteem jälgib ravimite koguseid.
FN-20	Süsteem saadab madala koguse teavitusi.
FN-21	Süsteem kuvab ravimite aegumiskuupäevi.
FN-22	Süsteem võimaldab ühekordset manustamist ilma raviskeemi lisamata.
FN-23	Süsteem võimaldab triipkoodi kaudu kontrollida, kas ravim kuulub profiili raviskeemi

Tabel 6. Mittefunktsionaalsed nõuded.

ID	Nõue
MFN-01	Põhifunktsioonid on kättesaadavad maksimaalselt kolme sammuga.
MFN-02	Veateated on arusaadavad ja sisaldavad tegevusjuhust.
MFN-03	Süsteem käsitleb tõrkeid, näiteks tundmatu triipkood, ilma rakenduse katkestusega.
MFN-04	Andmete salvestamine on järjepidev ja kontrollitav.
MFN-05	Kasutaja- ja ravimiandmetele juurdepääs on autentitud ja autoriseeritud.
MFN-06	Ravimiinfo pärineb Eesti riiklikust ravimiregistrist.
MFN-07	Rakendus töötab iOS ja Android operatsioonisüsteemidel
MFN-08	Süsteem on kihilise arhitektuuriga, mis toetab edasist arendust.
MFN-09	Andmeskeemi muutused on hallatud migratsioonidega.

Kasutajalugude ülesehitamisel lähtuti Cohni poolt defineeritud struktuurist „Kasutajana tahan [tegevus], et [äriline väärtus]” (Cohn, 2004, lk 81). Kasutajalood lisati GitLabi *issue*-dena, kus iga lugu tükeldati omakorda väiksemateks tehnilisteks ülesanneteks. See võimaldas arenduse käiku selgemalt jälgida ning tagada, et iga funktsionaalsus on lõpuks täielikult realiseeritud. Arenduse käigus loodi kokku 14 kasutajalugu, millest 13 sai lahendatud ning 55 väiksemat ülesannet, millest sai 54 lahendatud. Lisaks loodi kasutajalugude põhjal kasutusjuhu diagramm, mis on esitatud joonisel 1. Diagramm koostati põhimõttel, et saada parem ülevaade kasutajate seostest süsteemiga.



Joonis 1. MedScan kasutusjuhu diagramm.

4.1.3 Projekti seadistamine

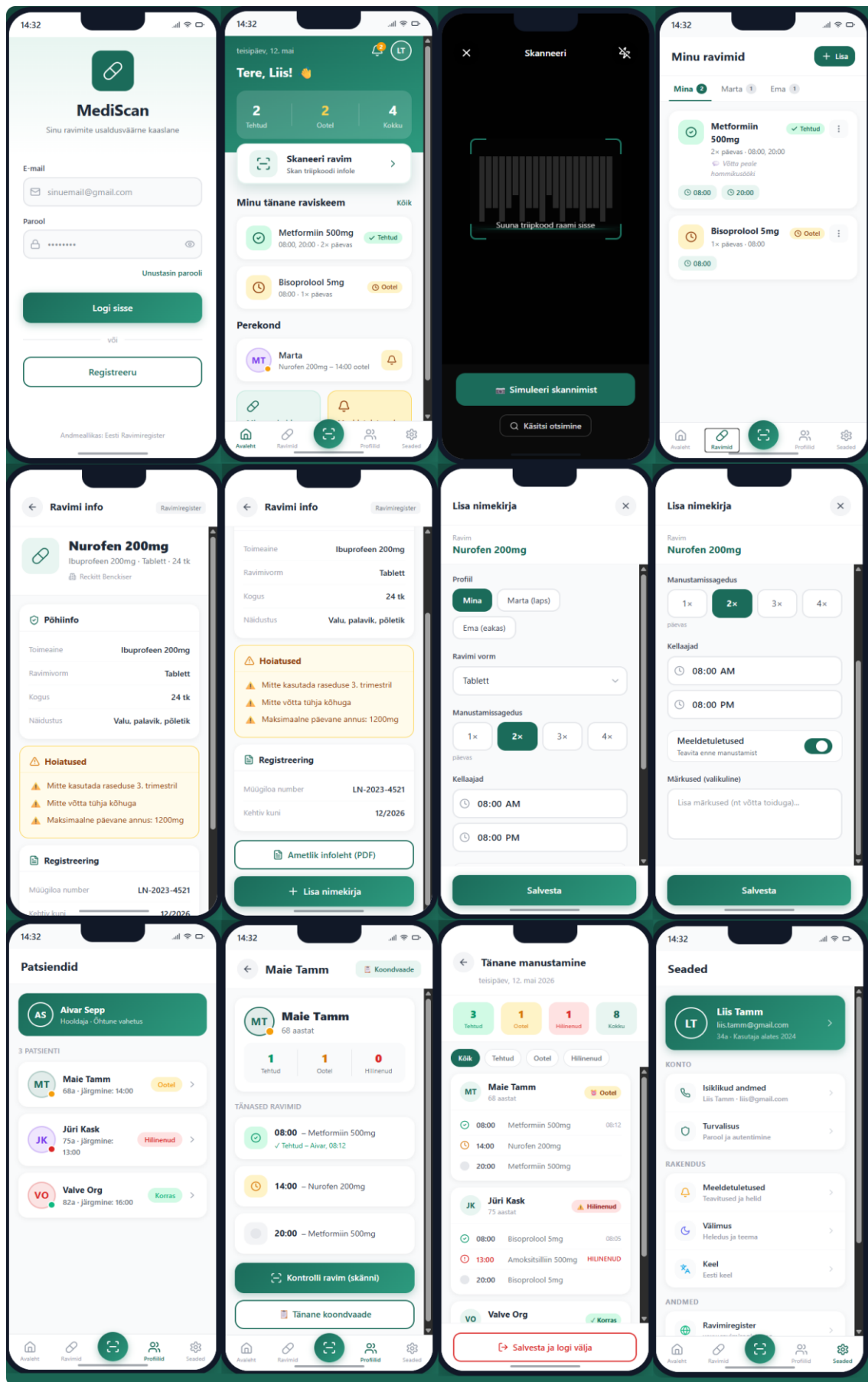
Pärast nõuete defineerimist alustati projekti tehnilise seadistamisega. Visual Studios loodi .NET MAUI Blazor Hybrid and Web Apps projekt, mis võimaldab ühe koodibaasi põhjal toetada nii mobiili- kui veebirakendust. Projekti struktuur oli alguses võõras ning autorid vajasisid aega mõistmaks, kus millised failid asuvad ja kuidas erinevad kihid omavahel suhtlevad. Pärast struktuuri selgeks saamist seadistati projekt nii, et mõlemad arendajad said korraga koodi kirjutada ja GitLabi kaudu oma muudatusi jagada.

Paralleelselt projekti seadistamisega loodi andmebaasi skeem PostgreSQL-is ning seadistati Docker-põhine arenduskeskkond, mis tagas, et mõlemal arendajal oli identne töökeskkond olenemata kasutatavast arvutist.

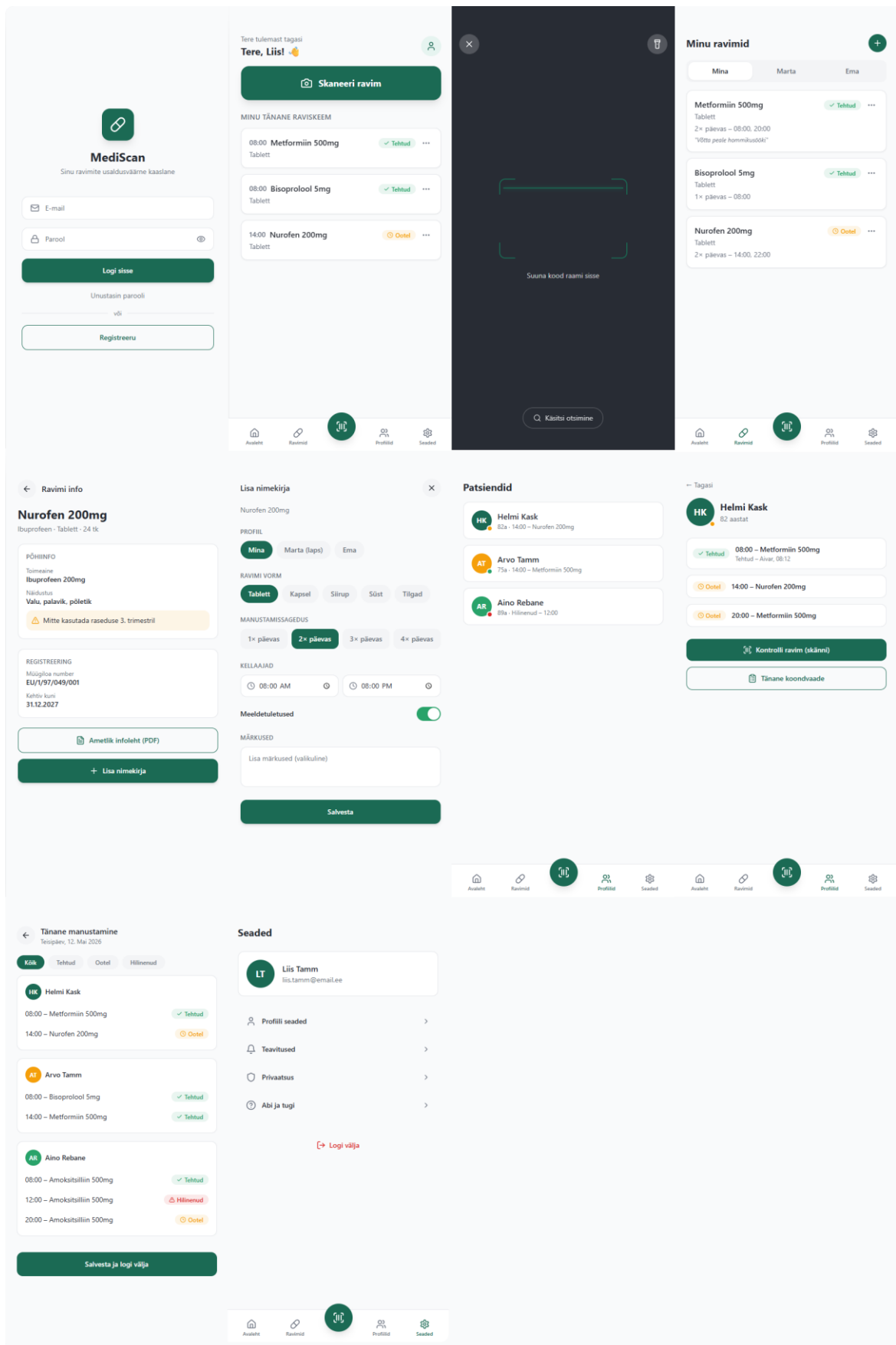
4.2 Rakenduse loomine

4.2.1 Figma ja Lovable prototüüpide loomine ja testimine

Enne päris rakenduse arendamist otsustati luua visuaalsed prototüübid, et kontrollida, kas sellisel rakendusel on kasutajate jaoks üldse praktilist väärtust. Persoonade, kasutajavoogude ja kasutajalugude põhjal koostati tehisintellektiassistent Claude (Anthropic, 2026) abil detailne *prompt*, mis kirjeldas rakenduse eesmärki, sihtkasutajaid ja vajalikke funktsionaalsusi. Valminud *prompt* sisestati disainitööriistadesse Figma Make (Figma, 2026) ja Lovable (Lovable, 2026), mis genereerisid automaatselt interaktiivsed visuaalsed prototüübid. Figma Make-is koostatud disain on joonisel 2 ning Lovable-is joonisel 3.



Joonis 2. Figma Make-is koostatud disain.



Joonis 3. Lovable-is koostatud disain.

Valminud prototüüpide põhjal viidi läbi eeltestimine, kus kuus potentsiaalset kasutajat said prototüüpidega tutvuda ning täita lühiküsitluse. Vastajate hulgas oli viis 18-29-aastast ning üks 45-60-aastane kasutaja. Neli vastajat kasutasid ravimeid aeg-ajalt ning kaks regulaarselt. Kõik kuus vastajat kasutasid nutitelefoni igapäevaselt väga sageli.

Küsitlus hõlmas küsimusi ravimiinfo leidmise, ravimite haldamise ning rakenduse funktsionaalsuste kohta. Ravimiinfo leidmise lihtsust hinnati skaalal 1–5, kus enim vastuseid koondus hinnetele 2 ja 3, mis näitab, et info leidmine on ebamugav. Peamiste probleemidena toodi välja, et info leidmine võtab liiga kaua aega, infoleht on sageli kadunud ning info on keerulises meditsiinikeeles. Kõige levinumad infootsimise viisid olid ravimi infoleht pakendis ja Google otsing ravimi nime järgi.

Ravimite haldamise osas selgus, et neli vastajat ei tea kõiki kodus olevaid ravimeid ning kaks vastajat ei kontrolli aegumistähtaegu üldse. Ravimite võtmiseks tugines viis vastajat mälule ning üks kasutas telefonimeeldetuletust.

Triipkoodipõhise tuvastamise kasulikkust hindas viis vastajat kõrgeima hindegga 5 ning üks hindegga 4, mis näitab tugevat huvi sellise funktsionaalsuse vastu. Kõige väärtuslikumate funktsionaalsustena toodi välja isiklik ravimite nimekiri, triipkoodi skaneerimine, lihtsustatud ravimi põhiinfo ning ravimi võtmise meeldetuletused. Kõik kuus vastajat pidasid skaneerimise kaudu ravimite lisamist palju mugavamaks võrreldes käsitsi otsimisega.

Tagasiside põhjal kinnitati algne funktsionaalsuste loend ning määratleti MVP, mis pakub kasutajale reaalselt lisaväärtust. MVP hulka kuulusid triipkoodi skaneerimine, ravimi põhiandmete kuvamine ning lihtne koduapteegi haldus. Kõik täiendavad funktsionaalsused, nagu raviskeemide ja erinevate profiilide haldus planeeriti lisada peale MVP valmimist.

4.2.2 Prototüübi loomine

Rakenduse arendamine jagunes meeskonnaliikmete vahel selgelt. Stella tegeles rohkem rakenduse *backend* funktsionaalsustega ning Rando *frontend*-osaga. Tööjaotus tagas, et mõlemad said keskenduda oma tugevustele ning töö kulges efektiivselt ilma kattuvusteta. Iga tööpäeva alguses lepitati lühidalt kokku, mida kumbki arendaja sellel päeval teeb, et vältida topelttöö tegemist. Peatükis 5.6 analüüsivad autorid oma panust lähemalt.

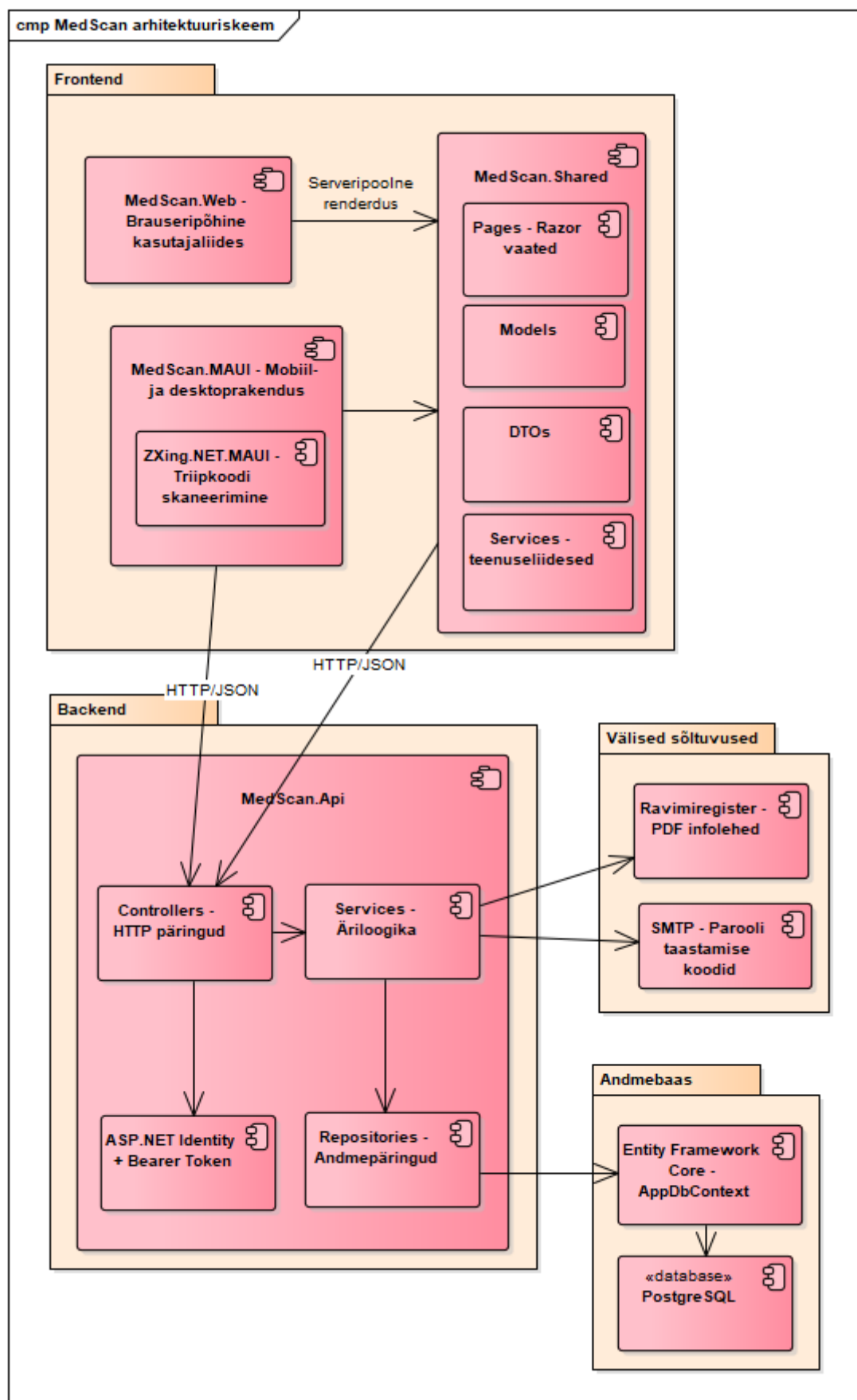
Arendus kulges etapiviisiliselt. Esmalt realiseeriti autentimise loogika, milleks kasutati ASP.NET Identity raamistikku. Seejärel loodi ravimite andmemudel ning CRUD-funktsionaalsus, implementeeriti triipkoodiskanner koos vastava API otsinguloogikaga. Järgnevalt arendati välja profiilide haldus koos hoolealuste lisamise võimalusega. Loodi seadete vaade ning korrastati rakenduse põhivaated. Pärast põhifunktsionaalsuste valmimist loodi Figmas ühtne tüpograafia ja kujundusjuhend, mille põhjal viidi kogu kasutajaliides visuaalselt ühtlaseks.

4.2.3 Rakenduse arhitektuur

MedScan on üles ehitatud kolmekihilise arhitektuuri põhimõttel. Rakendus koosneb kasutajaliidese kihist, serveri kihist ja andmebaasi kihist. Serveri kiht sisaldab nii API endpointe kui ka backend-i ärioloogikat ühes teenuses. Kontrollerid, teenused ja repository kiht on juurutatud samas MedScan.Api projektis. Selline jaotus eraldab kasutajaliidese, ärioloogika ning andmehalduse vastutused, parandades süsteemi hallatavust ja laiendatavust. Rakenduse rollid ja nende projektid on esitatud tabelis 7 ning arhitektuuriskeem on esitatud joonisel 4.

Tabel 7. MedScan rakenduse arhitektuur.

Projekt	Kirjeldus
MedScan.Api	<i>Backend</i> -i projekt, mis realiseerib API ning vastutab ärioloogika, autentimise, andmebaasiga suhtlemise ja väliste sõltuvustega seotud päringute eest.
MedScan.MAUI	<i>Frontend</i> -i projekt, mis on loodud .NET MAUI Blazor Hybrid raamistikus ning sisaldab mobiili- ja desktoprakenduse põhifunktsionaalsust, sealhulgas triipkoodi skaneerimist ZXing.NET.MAUI teegi abil.
MedScan.Shared	Sisaldab jagatud Razor vaateid, DTO-sid, mudeleid, teenusliideseid ja ühiseid osi, mida kasutavad rakenduse teised projektid.
MedScan.Web	Brauseripõhine kasutajaliides, mis kasutab serveripoolset renderdust ning on lahenduses olemas, kuid ei olnud töö käigus arenduse põhifookus.

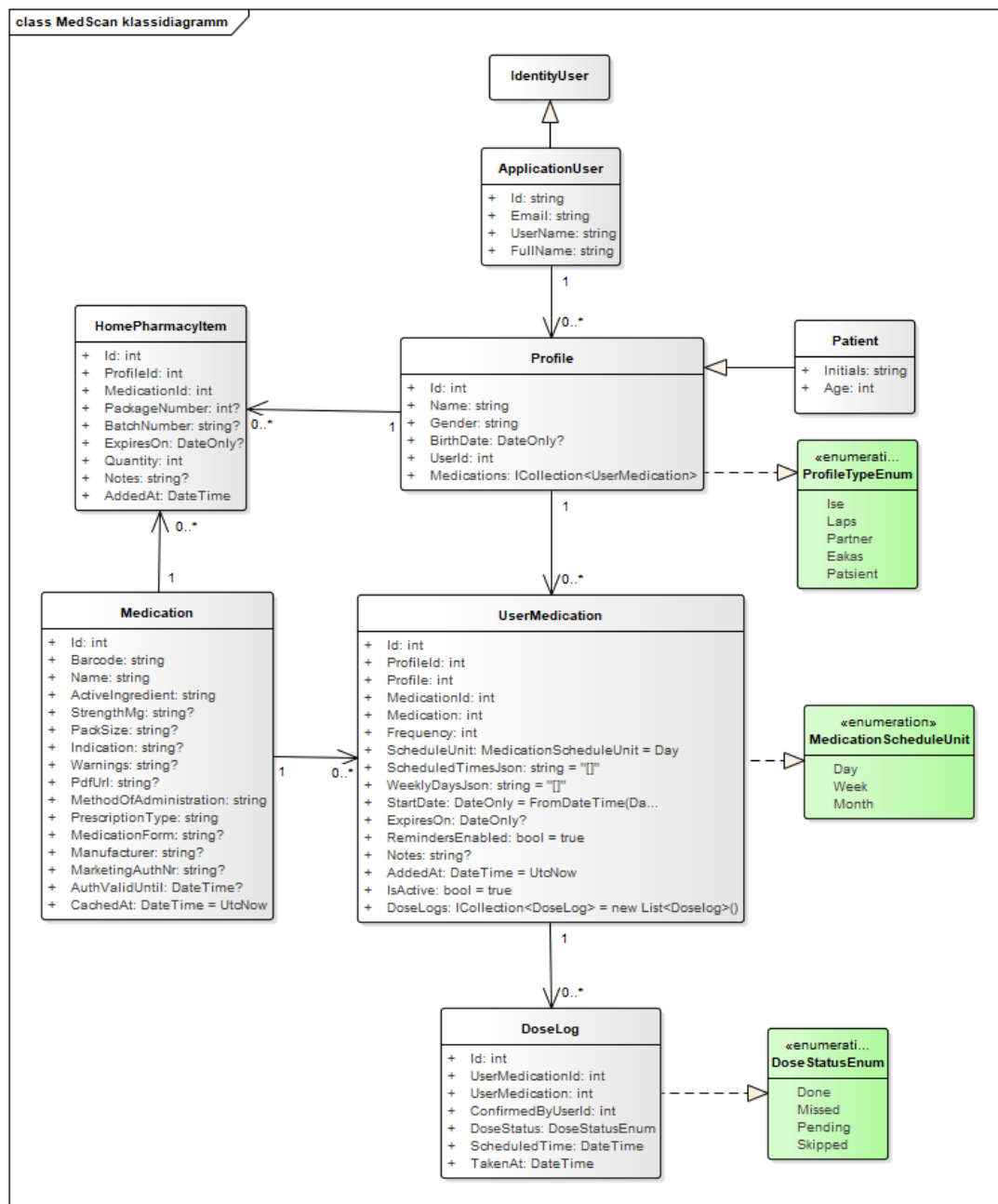


Joonis 4. MedScan süsteemi arhitektuuriskeem.

4.2.4 Andmemudel

Andmebaas on realiseeritud Entity Framework Core migratsioonidega, mis tagavad andmemudeli versioonihalduse ja muudatuste jälgitavuse. Joonisel 5 on esitatud rakenduse andmemudeli klassidiagramm, mis kujutab rakendusspetsiifiliste tabelite struktuuri ja nendevahelisi seoseid.

Andmemudeli keskmes on Profile klass, mis esindab nii kasutaja enda kui ka hoolealuste profile ning on seotud ApplicationUser'iga. Iga profiil sisaldab UserMedication kirjeid ehk raviskeeme, mis viitavad Medication klassile, kus hoitakse ravimi üldandmeid. Manustamise ajalugu salvestatakse DoseLog kirjetena, HomePharmacyItem klass aga haldab koduapteegi koguseid ja aegumiskuupäevi. ASP.NET Identity tabelid on diagrammil esindatud üldistatud klassiga IdentityUser, kuna Identity raamistik genereerib need automaatselt ning nende sisemine struktuur ei mõjuta rakenduse ärilooikat. Iga rakendusspetsiifiline tabel on diagrammil tähistatud eraldi klassina koos atribuutide ja andmetüüpidega.



Joonis 5. Klassidiagramm MedScan-i peamiste andmebaasimudelite kohta.

4.2.5 Prototüübi testimine

Pärast põhifunktsionaalsuste valmimist viidi läbi teine voor kasutajateste, kus kuus kasutajat täitsid etteantud kasutusstsenaariumid päris rakenduses. Kõik kuus kasutajat läbisid testid edukalt ning leidsid, et rakendus on lihtne ja arusaadav. Testitulemused koos leitud probleemide ja tehtud parandustega on esitatud tabelis 8.

Tabel 8. Järeldestamise tulemused.

Stsenaarium	Oodatud tulemus	Tegelik tulemus	Leitud probleem	Parandus
Registreerimine	Konto luuakse edukalt	Konto loodi edukalt	Paroolinõuded pole enne sisestamist nähtavad; soo valikul puudus "ei soovi avaldada" variant	Lisati paroolinõuete kuvamine; lisati kolmas valik
Registreerimine	Konto loomise järel suunatakse kohe rakendusse	Konto loodi, kuid suunati kõigepealt vahelehele	Pärast registreerimist kuvati eraldi vaheleht	Eemaldati ebavajalik vaheleht
Sisselogimine	Kasutaja suunatakse avalehele	Kasutaja suunati edukalt avalehele	-	-
Teavituste logi avamine	Teavituste modal avaneb ja sulgub korrektselt	Teavituste modal avanes ja sulges edukalt	-	-
Profiili seadete avamine ja andmete muutmine	Põhikasutaja seaded vaade avaneb ja andmeid saab muuta	Seadete vaade avanes edukalt ja andmed salvestati korrektselt	-	-
Profiilide vaate avamine	Profiilide vaade avaneb ja põhikasutaja kaart on nähtav	Profiilide vaade avanes ja põhikasutaja kaart kuvati edukalt	-	-
Triipkoodi skanneerimine	Ravim tuvastatakse ja kuvatakse ravimi infoleht	Ravim tuvastati ja ravimi infoleht avati edukalt	-	-
Ametliku infolehe avamine	Ravimi ametlik PDF avaneb ravimiregistris	PDF avati edukalt	-	-
Hoolealuse lisamine	Profiil luuakse ja kuvatakse eraldi sektsioonina	Profiil loodi ja kuvati eraldi sektsioonina	Mõiste "hoolealune" segane kasutajale	Asendati terminiga "hoolealune"
Hoolealuste andmete muutmine	Muudetud andmed salvestatakse edukalt	Andmed salvestati edukalt	-	-
Ravimi lisamine	Ravim lisatakse valitud profiilile	Ravim lisati edukalt, kuid osad kasutajad alustasid valest kohast	Hoolealusele ravimi lisamine tekitas segadust, lisati valest kohast	Täpsustati kasutajaliidese sõnastust
Ravimi lisamine	Kasutusviisi valik on selge	Kasutusviisi valik lisati raviskeemi edukalt	Nupu sõnastus oli ebaselge	Muudeti sõnastust selgemaks

Stsenaarium	Oodatud tulemus	Tegelik tulemus	Leitud probleem	Parandus
Ravimi lisamine	Pärast salvestamist suunatakse ravimite vaatesse	Salvestamise järel suunati automaatselt ravimite vaatesse	-	-
Ravimi märkuste muutmine	Märkused salvestatakse edukalt	Märkused salvestati edukalt	-	-
Ravimi koguse muutmine	Kogus salvestatakse edukalt	Kogus salvestati edukalt	-	-
Raviskeemi lisamine	Uus raviskeem salvestatakse edukalt	Uus raviskeem salvestati edukalt	-	-
Raviskeemi muutmine	Sagedus ja kellaajad salvestatakse edukalt	Sagedus ja kellaajad salvestati edukalt	-	-
Raviskeemi lõpetamine	Raviskeem lõpetatakse edukalt, kuvatakse kinnitus	Raviskeem lõpetati edukalt ja kasutajale kuvati kinnitus	-	-
Manustamise kinnitamine	Manustamine kinnitatakse ühe hüplikaknega, hüplikakna tekstid on selged	Manustamine kinnitati edukalt, tekstid olid selged, kuid kuvati kaks hüplikakent korraga	Varajase manustamise kinnituse hüplikaken ja madala koguse hoiatus kuvati samaaegselt	Kõrvaldati topelt hüplikaken
Manustamis-ajaloo vaatamine	Ajalugu kuvatakse korrektselt	Ajalugu kuvati korrektselt	Ajavahemiku valik puudus, sai valida ainult konkreetse kuupäeva	Planeeritud edasiseks arenduseks
Meeldetuletused	Teavitused kuvatakse teavituste logis	Teavitusi ei kuvatud teavituste logis	Teavituste logi ei kuvatud <i>bell</i> -nupu juures	Parandatud
Seadete avamine	Kõik seadete vaated avanevad korrektselt	Kõik seadete vaated avanesid korrektselt	-	-
Väljalogimine	Kasutajale kuvatakse eduka väljalogimise teade ning ta suunatakse sisselogimise vaatesse tagasi	Kasutajale kuvati eduka väljalogimise teade ning ta suunati sisselogimise vaatesse tagasi	-	-

Positiivsena toodi välja ravimi lisamise lihtsus ja loogilisus, manustamise logi kasulikkus ning ravimite koguse ja aegumiskuupäevade jälgimine. Saadud tagasiside põhjal tehti

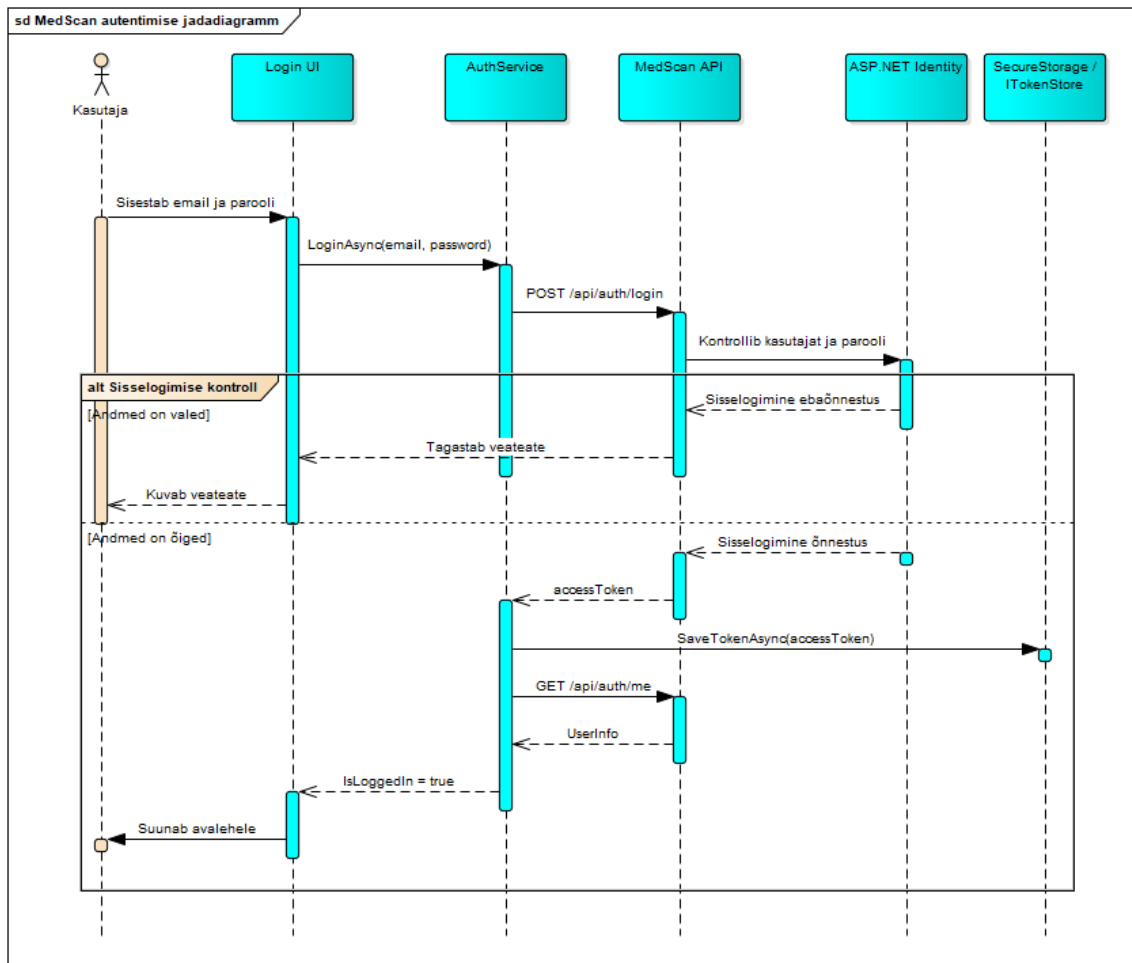
rakenduses vastavad parandused ning täiendavad soovitusel, mille realiseerimine ei mahtunud käesoleva töö raamidesse, on koondatud peatükki 5.5.

4.3 Realiseeritud funktsionaalsused

4.3.1 Autentimine

Uus kasutaja loob konto e-posti aadressi ja parooliga. Registreerimisvormil tuleb sisestada nimi, e-posti aadress, parool, sugu ja sünnikuupäev. Edukal registreerimisel suunatakse kasutaja automaatselt rakendusse. Kui sisestatud e-posti aadressiga konto on juba olemas, kuvatakse vastava sisuga veateade.

Olemasolev kasutaja logib sisse e-posti aadressi ja parooliga. Eduka autentimise järel väljastab server JWT-i, mis salvestatakse seadmesse ning lisatakse automaatselt kõigile järgnevatele API päringutele. See tagab, et kasutaja ei pea iga rakenduse avamisel uuesti sisse logima. Vale e-posti aadressi või parooli sisestamisel kuvatakse selge veateade ilma täpsustamata, kumb väli on vale. See on teadlik turvavalik, mis ei lase pahaloomulisel kasutajal järeldada, kas konkreetne e-posti aadress on süsteemis registreeritud. Prototüübi autentimise protsess on kujutatud joonisel 6.

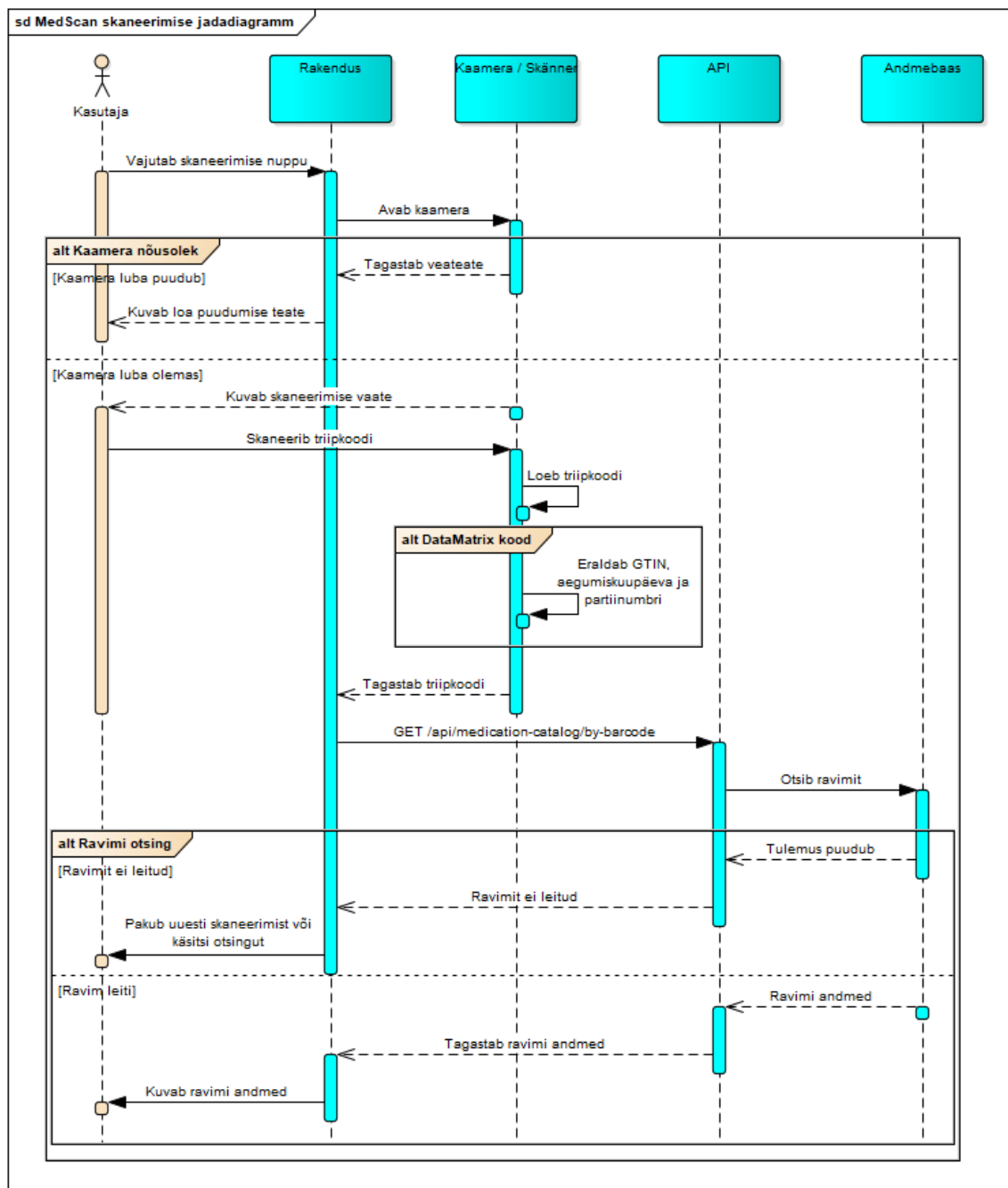


Joonis 6. Autentimise jadadiagramm

Kui kasutaja on parooli unustanud, saab ta sisselogimisvaatelt avada parooli muutmise voo. Kasutaja sisestab oma e-posti aadressi ning rakendus saadab meilile 6-kohalise koodi. Kui see kood rakendusse sisestada, saab kasutaja uuendada oma parooli. Pärast edukat parooli muutmist suunatakse kasutaja tagasi sisselogimisvaatesse. Prototüübi parooli muutmise protsess on kujutatud lisas 3, joonisel 12.

4.3.2 Triipkoodi skaneerimine ja ravimi tuvastamine

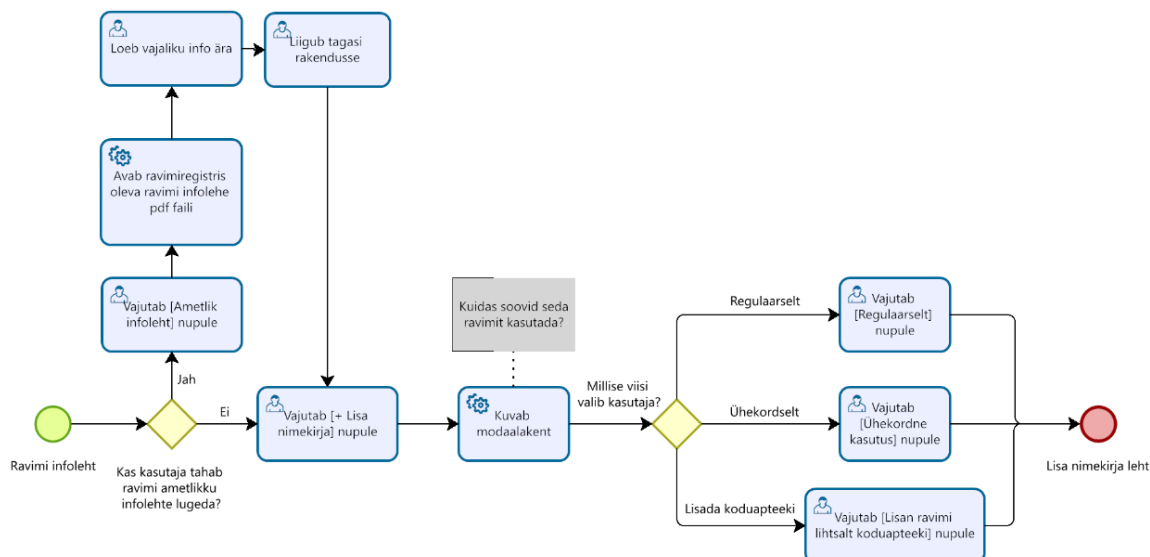
Rakenduse avaleht sisaldab nuppu „Skaneeri ravim“. Nupule vajutades küsib rakendus kaameraõigust ning avab skännervaate. Skanner tuvastab triipkoodi automaatselt ilma täiendava nupuvajutuseta ning toetab nii EAN-13 kui DataMatrix formaate. Kehva valgustuse korral võimaldab rakendus taskulambi sisselülitamist. Pärast edukat tuvastamist otsitakse ravim andmebaasist GTIN-koodi alusel. Kui ravimit ei leita, kuvatakse teade „Tundmatu triipkood“ ning pakutakse käsitsi ravimi nime järgi otsimise võimalust. Ravimi skaneerimise protsess on kujutatud joonisel 7 ning skaneerimise loogika kuvatõmmised on esitatud töö lõpus lisas 3, joonisel 13.



Joonis 7. Skaneerimise jadadiagramm.

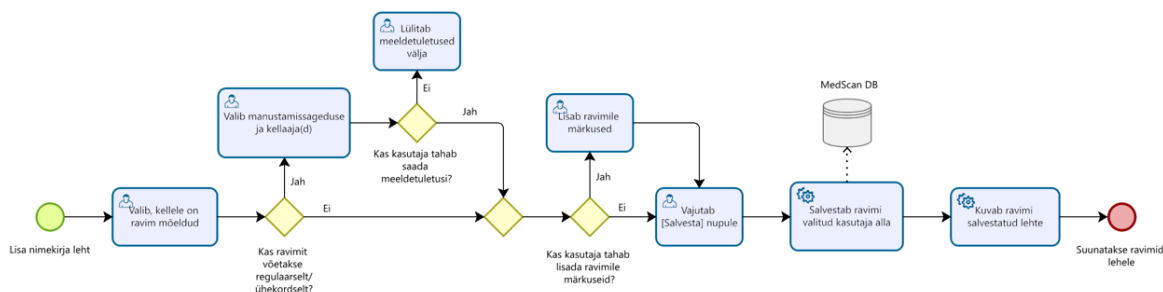
4.3.3 Ravimi infoleht

Infolehel kuvatakse ravimi nimi, toimeaine, ravimivorm, tugevus, kogus, näidustus, aegumiskuupäev ning peamised hoiatused. Kuvatavad andmed pärinevad rakenduse andmebaasist, mis on täidetud ravimiregistri andmete põhjal. Lisaks on lehel nupp „Ametlik infoleht“, mis avab ravimiregistris vastava PDF-dokumendi. Sealt leiab kasutaja täieliku ravimiteabe. Ravimi infolehe avamise protsess on kujutatud joonisel 8 ja infolehe kuvatõmmised on esitatud lisas 3, joonisel 14.



Joonis 8. Ravimi infolehe BPMN-diagramm.

Kasutaja saab lisada ravimeid koduapteeki nii skaneerimise kui nime järgi otsimise teel. Ravimi lisamisel määratakse kogus automaatselt pakendi suuruse põhjal. Koduapteegis kuvatakse kõik ravimid koos koguste ja aegumiskuupäevadega. Kogus väheneb automaatselt iga manustamise märkimisel. Ravimi lisamise protsess on kujutatud joonisel 9.



Joonis 9. Ravimi lisamise BPMN-diagramm.

4.3.4 Raviskeem ja meeldetuletused

Kasutaja saab ravimeid lisada raviskeemi läbi koduapteegi ehk ravimite lehe või skanneerimise teel. Lisades määrab kasutada ravimi võtmise sageduse, kellaajad ja meeldetuletuse nõusoleku. Rakendus saadab teavituse kujul „[profiilinimi] - aeg võtta [ravimi nimi]”. Teavituse ilmunisel saab kasutaja valida, kas vajutada „Võetud” nupule või sulgeda teavitus. Ravimi märkimisel võetaks logitakse kella-aeg ja kuupäev. Raviskeemi lisamise lehed on esitatud lisas 3, joonisel 15 ning meeldetuletuste vaated joonisel 16.

Kasutaja saab märkida ravimi ühekordselt võetuks, ilma et see lisanduks igapäevasesse raviskeemi. See on mõeldud olukordadeks, kus ravimit tarbitakse vaid ühekordselt, näiteks valuvaigisti või palavikuravim. Koduapteegis oleva ravimi kogus väheneb automaatselt ja toiming salvestatakse manustamise logisse.

4.3.5 Madala koguse teavitused

Rakendus jälgib ravimite koguseid ning saadab teavituse, kui ravimi kogus jõuab viimase viie doosini. Teavitus saadetakse iga järgmise doosi võtmisel kuni jõutakse viimase manustamiseni. Madala koguse teavituse kuvatõmmis on esitatud lisas 3, joonisel 17.

Kasutaja saab luua oma konto alla ühe või mitu hoolealuse profiili. See on eriti kasulik olukordades, kus üks pereliige vastutab teiste, näiteks laste või eakate vanemate raviskeemide järgimise eest.

Uue hoolealuse profiili loomisel sisestatakse nimi, sugu ja sünnikuupäev. Pärast loomist on hoolealuse profiil koduapteegi vaates nähtav eraldi sektsioonina, mis hoiab eri inimeste ravimid selgelt lahus. Igale hoolealusele saab lisada oma ravimid, seadistada raviskeemi ning vaadata manustamise logi, mis annab ülevaate sellest, kas ja millal ravimeid on võetud. Hoolealuse andmeid saab igal ajal muuta. Vajadusel saab hoolealuse profiili ka kustutada, misjärel eemaldatakse tema andmed kasutaja kontolt.

Lisaks on võimalik ravimit triipkoodi abil kontrollida. Rakendus tuvastab, kas skaneeritud ravim kuulub just selle hoolealuse raviskeemi. See on praktiline abivahend olukordades, kus kodus on mitu erinevat ravimit ning tuleb kiiresti veenduda, et õige inimene saab õige ravimi. Hoolealuste lehe haldamise kuvatõmmised on esitatud lisas 3, joonisel 18.

5 Analüüs ja järeldused

5.1 Tehnilise teostuse põhjendus

Tehnoloogiate ja tööriistade valikul lähtuti kolmest põhimõttest: juhendajate soovitudest, autorite varasemast kogemusest ning konkreetse vahendi sobivusest lahendatava ülesande jaoks.

Põhiraamistikuks valiti .NET MAUI Blazor Hybrid juhendajate Dr Gunnar Piho ja Viljam Puusepa soovitusel. Andmebaasikihis kasutati PostgreSQL-i koos Docker-ga, kuna mõlemad olid autoritele tuttavad ning konteineriseerimine tagas arendus- ja toodangukeskkondade ühtsuse. Autentimise realiseerimiseks valiti ASP.NET Identity, kuna see on raamistikku sisse ehitatud. Vajalikud tabelid ja loogika genereeritakse selles raamistikus automaatselt ning see välistas topelttöö. Triipkoodimoodulina kasutati ZXing teeki, kuna see toetas nii EAN-13 kui ka DataMatrix formaate ning see integreerus projekti sujuvalt.

Kujunduse ja tüpograafia loomiseks valiti Figma, millega mõlemad autorid olid varasemalt töötanud. See võimaldas luua ühtse visuaalse stiili, millest lähtuti kogu arenduse vältel. Diagrammide koostamiseks kasutati Enterprise Architect-i ja Bizagi Modeler-it, mis olid autoritele tuttavad varasematest õppekava kursustest. Versioonihalduseks valiti TalTech GitLab. See võimaldas kahel arendajal mugavalt ühe koodibaasi kallal paralleelselt töötada. Arendusprotsessi kiirendamiseks võeti kasutusele Codex ja GitHub Copilot, mis vähendasid korduvkoodi kirjutamist ning tänu millele oli võimalus keskenduda keerulisemate probleemide lahendamisele.

Koodi kirjutamisel järgiti *Clean Code* põhimõtteid, mida rõhutati ka õppekava kursustel. Põhireeglid olid:

- Meetodid täidavad ühte selgelt määratletud ülesannet.
- Muutujate ja klasside nimed on kirjeldavad.
- Koodi struktuur on loetav ilma lisakommentaarideta.

Nimetatud põhimõtteid püüti järgida kogu arenduse vältel, kuid töö mahu ja ajalise piirangu tõttu ei õnnestunud neid kõikjal täielikult rakendada. Arendusfaasi lõpus viidi läbi osaline koodi korrastus, kuid projektis esineb siiani duplikaatkoodi ning täielikku refatoreerimist ei jõutud teostada.

5.2 Hinnang projekti teostamise protsessi kohta

Projekti arendus kestis ligikaudu neli kuud ning selle jooksul tehti kokku 201 commiti. Meeskond koosnes kahest arendajast. Broneeriti TalTech õpiruum minimaalselt kolm korda nädalas, kus töötati keskmiselt neli kuni kuus tundi korraga. Lisaks ühisele tööajale töötas kumbki arendaja ka individuaalselt.

Kohapealne koostöö osutus väga tõhusaks. Küsimuste tekkimisel saadi kohe nõu küsida ning tagasisidet anda. Kriitika vastuvõtmine ja ühine otsuste tegemine kulges sujuvalt, kuna mõlemad autorid olid ülikooli esimesest aastast alates erinevates paarisprojektides koos töötanud ning teadsid, et nende suhtlus- ja töötamisviis klapib. Erimeelsuste korral analüüsiti probleemi ühiselt ning leiti mõlemale sobiv lahendus. Suuri arusaamatusi ei esinenud.

Paarisarenduse formaat tõestas oma väärtust mitmel tasandil. Õppejõud on korduvalt maininud, et paaristöö toob kaasa parema koodi kvaliteedi ja vähem vigu võrreldes individuaalse arendusega. Praktika kinnitas seda ka käesolevas projektis. Kahekesi suudeti lühikese ajaperioodi jooksul valmis teha sisukam tulemus, kui seda oleks individuaalselt plaanitud teha. Lisaks hoidis ühine tööritm motivatsiooni üleval, kuna mõlemal oli alati võimalus teist liiget suunata ja innustada.

Arenduse käigus esines üks katkestus. Aprilli alguses tehti nädalane paus, mille järel võttis projekti uuesti sisseelamine pisut aega. See kogemus näitas, et järjepidevus on iteratiivses arenduses kriitilise tähtsusega ning pikemad pausid võivad oluliselt aeglustada tööde edenemist isegi siis, kui tehniline koodibaas on puutumata.

5.3 Projekti kitsaskohad ja õnnestumised

Projekti suurimaks õnnestumiseks võib pidada seda, et kõik MVP funktsionaalsused said realiseeritud ning kasutajatestid kinnitasid, et rakendus pakub reaalselt lisaväärtust.

Negatiivne tagasiside puudutas vaid väiksemaid kasutajaliidese parandusi, mis said kõik korrastatud. Tööriistade kasutamisel olulisi takistusi ei esinenud ning juhendajatega sujus suhtlus hästi. Vajadusel saadi kiiresti abi.

Kuna teema oli autorite poolt ise valitud, püsis huvi ja motivatsioon kogu projekti vältel stabiilsena. Meeskonnasiseselt jaotus töö võrdselt ning mõlemad liikmed said tegeleda valdkondadega, milles nad end kõige tugevamalt tundsid. Kordagi ei tekkinud tunnet, et panus oleks ebaõiglaselt jaotunud.

Peamiseks kitsaskohaks kujunes andmeallika puudumine. Algselt planeeriti kasutada ravimiregistri veebilehelt allalaaditavat CSV-faili (Ravimiamet, i.a.-b), kuid analüüs näitas, et seal puuduvad EAN-13 ja DataMatrix koodide andmed. Kuna triipkoodipõhine tuvastamine on rakenduse peamine funktsionaalsus, muutis see CSV-faili kasutuskõlbmatuks. Selle tulemusena sisestati funktsionaalsuse demonstreerimiseks ja testimiseks 16 ravimit rakenduse andmebaasi. Andmebaasis olevate ravimite pakendid ja triipkoodid on esitatud lisa 2, joonistel 10 ja 11.

Samuti osutus kitsaskohaks teema režiimi funktsionaalsus. See implementeeriti ning sellele pühendati arvestatav hulk arendusaega, kuid hilisema UI korrastamise käigus läks see katki. Ajapuuduse tõttu otsustati see algsest versioonist välja jätta ning korrektselt realiseerida tulevases arendusetapis.

5.4 Lahenduse äriline ja praktiline põhjendatus

MedScan lahendab reaalse probleemi, mida kinnitas nii eelküsitlus kui ka järeltestimine. Kasutajatel on vaja kiiret juurdepääsu usaldusväärsele ravimiinfole ning mugavat koduapteegi haldust. Rakendus on potentsiaalselt relevantne laiale sihtrühmale. Igapäevaselt ravimeid kasutavatele täiskasvanutele, hooldajatele ning peredele, kes haldavad mitme inimese raviskeeme.

Ärilisest vaatepunktist on mobiilne tervisetehnoloogia kiiresti kasvav sektor. Rahvusvahelised uuringud näitavad, et tervise- ja ravimirakenduste turg kasvab aasta-aastalt ning kasutajate valmisolek selliste lahenduste kasutamiseks suureneb. Eesti kontekstis toetab lahenduse asjakohasust riigi üldine digitervise infrastruktuur ning elanikkonna kõrge digipädevus.

Rakenduse potentsiaalsed tulumudeli võimalused hõlmaksid näiteks tasuta põhiversiooni koos tasulise lisafunktsionaalsusega, institutsionaalset litsentsimist haiglate või hooldekodude jaoks või integratsiooni apteekide infosüsteemidega. Samas tuleb arvestada, et tervisevaldkonna rakenduste kommertsialiseerimisega kaasnevad rangemad regulatiivsed nõuded, andmekaitse kohustused ning vajadus meditsiinilase sertifitseerimise järele, mis eeldavad täiendavat investeeringut.

Tuleb siiski rõhutada, et tehtud töö raames ei ole läbi viidud põhjalikku ärianalüüsi ega tasuvusuuringut. Esitatud hinnangud on kontseptuaalsel tasandil ning reaalse ärilise potentsiaali hindamine eeldaks eraldi turu-uuringut, konkurentsianalüüsi ning finantsmodelleerimist.

5.5 Projekti tulevikuplaanid

Prototüübi valmimine on alles lähtepunkt. Rakenduse edasiarendamiseks on mitmeid sisukaid võimalusi, mis muudaksid selle veelgi kasulikumaks ja laiemalt kättesaadavamaks.

Esimese sammuna on plaanis koodi kvaliteedi tõstmine. Koodibaasis esinevad duplikaatklassid refaktoreeritakse ning *Clean Code* põhimõtted rakendatakse järjepidevalt kogu projekti ulatuses. Refaktoreerimine on edasiarenduste eelduseks, kuna puhas ja hooldatav koodibaas vähendab uute funktsionaalsuste lisamise aega.

Täielik ravimiregistri integratsioon on funktsionaalsuse seisukohalt järgmine kõige kriitilisem edasiarendus. Hetkel sisaldab andmebaas 16 käsitsi sisestatud ravimit, kuid ideaalse lahendusena võiks rakendus olla ühendatud ravimiregistri täieliku andmebaasiga koos triipkoodide andmetega. See muudaks rakenduse igapäevaselt kasutatavaks ning kõrvaldaks peamise piirangu, mis takistab praegu laialdasemat kasutust.

Sellega seonduvalt on plaanis lisada ka sisselogimist mittenõudev ravimiinfo kiirotsing. Kasutaja saaks ravimi triipkoodi skaneerides koheselt kätte pakendi infolehe ja põhiandmed, ilma et ta peaks kontot looma või oma ravimite nimekirja haldama. See toetaks olukorda, kus inimene soovib kiiresti konkreetse ravimi kohta infot saada.

Smart-ID integratsioon võimaldaks kasutajatel rakendusesse sisse logida ilma eraldi kontot loomata, kasutades olemasolevat digitaalset identiteeti. See tõstaks

usaldusväärst, kuna kasutaja isik oleks automaatselt tuvastatud. Smart-ID integratsioon on tehniliselt teostatav ning mitmete terviserakenduste poolt juba kasutusel.

Kasutajate omavaheline ühendamine lahendaks olukorra, kus pereliikmete ravimite haldamine toimuks eraldi profiilide all. Tulevikus võiks olla võimalik luua peregrupid, kus kõik liikmed näevad ühist koduapteeki ning saavad jälgida, kas teised on oma ravimeid võtnud. See on eriti oluline eakate ja krooniliste haigustega patsientide hooldamisel.

Perearstide vaade võimaldaks arstidel näha patsiendi ravimiandmeid otse rakendusest. See parandaks ravikoostööd ja vähendaks infokadu. Selline integratsioon eeldaks aga tervise infosüsteemidega ühendamist. Samuti oleks vaja rangemat andmekaitsealast regulatsiooni, mis teeb sellest keerukama ja kulukama arenduseesmärgi.

Kasutaja personaliseerimine tähendaks, et kasutaja saab sisestada oma vanuse, kaalu, kroonilised haigused nagu diabeet jne. Rakendus kohandaks sellest lähtuvalt hoiatusi ja annuste soovitusi. Selline funktsionaalsus tõstaks rakenduse väärtust märkimisväärselt. Samas eeldab see meditsiinalast valideerimist ning tõenäoliselt ka regulatiivset sertifitseerimist.

Ravimite tellimine võimaldaks kasutajal rakendusest otse apteeki tellimuse esitada, kui ravim hakkab otsa saama. Integratsioon apteekide logistikasüsteemidega on tehniliselt teostatav ning looks selge ärilise mudeli partnerluseks apteekidega.

Personaalne AI-assistent võimaldaks kasutajal kirjeldada oma sümptomeid ning saada soovitusi sobivate käsimüügiravimite kohta. Selline funktsionaalsus on tehnoloogiliselt üha kättesaadavam tänu suurte keelemudelite arengule. Meditsiini valdkonnas tuleb arvestada vastutusküsimuste ning regulatiivsete piirangutega.

Keele- ja hääle-API integratsioon lisaks rakendusele tekst-kõneks funktsionaalsuse, mis loeks ravimiinfo kasutajale ette. See oleks eriti väärtuslik nägemispuudega kasutajatele ning eakatele, kellele ekraanilt lugemine võib olla keeruline. Keelevalikuna oleks mõistlik toetada lisaks eesti keelele ka vene ja inglise keelt, arvestades Eesti mitmekeelset elanikkonda.

Toidulisandite tugi laiendaks rakenduse kasutusala ravimitelt vitamiinidele ja toidulisanditele nagu C-vitamiin, D-vitamiin ja kreatiin, mida paljud kasutajad võtavad regulaarselt.

Väiksematest täiendustest on plaanis realiseerida mitu kasutuskogemust parandavat muudatust. Uutele kasutajatele lisatakse sissejuhatav juhendvoog, mille saab soovi korral vahele jätta. Samuti on eesmärgiks lisada meeldetuletuste edasilükkamise funktsioon, mis võimaldaks teavitust hilisemaks edasi lükata. Tume režiim, mis oli prototüübis osaliselt implementeeritud, kuid UI korrastuse käigus katki läks, taastatakse täisfunktsionaalsena.

5.6 Autorite eneseanalüüs

5.6.1 Stella eneseanalüüs

Projektis täitsin peamiselt arendaja rolli. Minu vastutusalaks kujunes rakenduse tehniline selgroog.

Projekti alfaasis rakendasin autentimise loogika, mis hõlmab kasutaja registreerimist, sisselogimist ning JWT-põhist sessiooni haldust. Andmebaasi poolel alustasin SQLite-iga, kuid hiljem migreerisin andmebaasi PostgreSQL peale, et rakendus oleks võimalik konteineriseerida Docker-ga ning käivitata ühtses keskkonnas. Lisaks lõin kõigi põhiklasside CRUD operatsioonid, sealhulgas ravimite, raviskeemide ja koduapteegi halduseks.

Skaneerimisloogika implementeerimine hõlmas nii EAN-13 kui ka DataMatrix triipkoodide tuvastamist, kaamera loogikat ning skaneerimistulemuste tötlust. Arenduse käigus refaktoreerisin skannerivoo teenust, lisasin raviskeemi toimingute loogika ning vältisin duplikaatskaneerimist. Kõigi põhifunktsionaalsuste juures vastutasin ka *frontend*-i ja *backend*-i vaheliste ühenduste loomise eest.

Meeldetuletuste loogika osutus kõige ajamahukamaks komponendiks. See nõudis korduvat itereerimist enne töökindla lahenduseni jõudmist.

Seadete funktsionaalsuse raames tegin aktiivseks välimuse ja keele seaded. Lisasin ka navigeerimise ravimiregistri lehele ning kontaktinfo kuvamise. Implementeerisin tumeda režiimi, kuid UI lõppviimistluse käigus läks see katki. Otsustasime selle hetkel

eemaldada. Arendusfaasi lõpus tegin ulatusliku kujunduse korrastuse ning tõstsin `.razor.css` klassid eraldi failidesse, mis parandas koodi struktureeritust.

Testimise poole pealt koostasın eeltestimiseks Google Forms küsimustiku ning pärast arendust järeltestimise stsenaariumid. Mõlemad testietapid viisime läbi koos kaasarendajaga, kaasates tuttavaid vabatahtlike testijatena.

Vaadates tagasi projekti kulgemisele, oli suurimaks väljakutseks avaliku andmeallika puudumine. Otsustasime ravimid käsitsi rakenduse andmebaasi sisestada. Lisasin 16 testravimit, mis võimaldasid funktsionaalsust demonstreerida ja testida.

Projekti käigus arendasin võrdselt nii tehnilisi kui ka üldiseid arendusoskusi. Õppisin, kuidas triipkoodid toimivad ja kuidas neid ravimite puhul kasutatakse ning sain palju kindlust `.NET` koodi struktuuri ja Clean Code põhimõtete rakendamises. Lisaks sain esmakordselt põhjalikumalt kasutada OpenAI Codex tööriista, millega varasem kokkupuude oli olnud minimaalne. Avastasin, et see on väga kasulik abivahend keerulisemate funktsionaalsuste realiseerimisel ning märkasın selget arengut promptide kirjutamises. Mida täpsemalt probleemi kirjeldasin, seda puhtama tulemuse tööriist genereeris. Arenes ka oskus iseseisvalt infot otsida ja leitud teadmisi rakendada. Koolis omandatud teadmised toetasid projekti elluviimist hästi. Olemas oli teoreetiline alus, kuid projekt andis võimaluse neid teadmisi iseseisvalt päriselt rakendada ja süvendada.

Olen meie projekti lõpptulemusega rahul, sest kõik minu vastutusalas olevad põhifunktsionaalsused said realiseeritud ning töötavad stabiilselt. Samuti hindan positiivselt meie koostöö kujunemist. Töötasime mõlemad pühendunult ühise eesmärgi nimel, sest uskusime, et meie rakendusel on potentsiaali teha paljude inimeste elu lihtsamaks.

5.6.2 Rando eneseanalüüs

Projektis täitsın ma suurema osa ajast analüütiku rolli, kuid puutusin kokku ka projekti tehnilise poolega.

Algfaasis koostasın neli kasutajapersoonat, mille põhjal defineerisin kolm peamist kasutajavoogu, mis olid prototüübi loomises kõige olulisemad. Persoonade ja kasutajavoogude põhjal koostasın GitLabi kasutajalood ning tükeldasin need omakorda väiksemateks ülesanneteks, et arenduse käigus oleks selgelt jälgitav, millal üks *issue* on

lahendatud. Kasutajalugude põhjal koostas ka kasutusjuhu diagrammi, mis andis hea ülevaate süsteemi ja kasutajate vahelistest seostest.

Eeltestimises kasutatud visuaalsete prototüüpide loomiseks koondas ma persoonad, kasutajavood, kasutajalood ning kasutusjuhu diagrammi tehisintellekti Claude abil ühtseks *prompt*-iks ning sisestasin Figma Make-i ja Lovable-isse. Lisaks aitasin kaasa ka küsimustiku koostamisel ning viisin läbi eel- ja järeltestimised.

Rakenduse arenduse jooksul tegelesin ma loogika väljamõtlemisega ja lisa funktsionaalsuste arendamisega, näiteks parooli uuendamise loogika. Iga valminud funktsionaalsuse vaatasime üheskoos üle ja tegime vajalikud parandused.

Arenduse lõpufaasis lõin rakenduse põhifunktsionaalsuste UML diagrammid. Lisaks märkasin, et rakenduse tüpograafia oli ebaühtlane ning otsustasin Figma koostada ühtse tüpograafilise süsteemi. Ühtse tüpograafia tulemusel nägi rakenduse kasutajaliides palju professionaalsem välja.

Projekti väljakutseteks osutus minul rakenduse loogika välja mõtlemine. Aru saada, milliseid funktsionaalsusi päriselt meie prototüüpi lisada ning kuidas rakendus teha võimalikult kasutajasõbralikuks, et iga üks oleks suuteline seda kasutama ilma kellegi abita. Tagantjärele vaadates oleks pidanud kohe peale eeltestimist looma UML diagrammid ja ühise tüpograafia. See oleks säästnud nii minu kui ka kaasarendaja aega ning vältinud topelttööd.

Projekt käigus arendasin kõige rohkem enda analüütilist mõtlemist. Õppisin, millised on arenduse esimesed sammud, mida tuleb kindlasti alguses paika panna ja mida võib arenduses edasi lükata. Kuna teema oli minu enda välja mõeldud, püsis motivatsioon kogu projekti vältel kõrgena. Olen meie tulemusega väga rahul, sest rakendus näeb väga professionaalne välja ning kõik põhifunktsionaalsused said prototüüpi integreeritud. Samuti olen rahul oma kaasarendaja valikuga. Koostöö sujus meil hästi ning töötasime mõlemad projekti kallal, kuna teadsime, et sellel lahendusel on potentsiaali aidata väga paljusid inimesi.

6 Kokkuvõte

Käesolev töö käsitleb triipkoodipõhise ravimihaldusrakenduse kavandamist ja arendamist Eesti kontekstis. Töö eesmärk oli analüüsida selle vajadust ning arendada prototüüpne mobiilirakendus MedScan, mis võimaldab triipkoodipõhist ravimituvastust ja isikliku raviskeemi haldamist. Töö metoodiliseks aluseks võeti Design Science lähenemine, mille kohaselt omandatakse teadmised probleemi kohta rakenduse kavandamise ja arendamise käigus. Kasutajavajaduste kaardistamiseks loodi persoonad ja kasutajalood ning viidi läbi kaks vooru kasutajatest. Eeltestimine visuaalsete prototüüpide põhjal ja järeltestimine valminud rakendusega.

Tulemusena valmis .NET MAUI Blazor Hybrid raamistikus mobiilirakendus, mis toetab nii EAN-13 kui DataMatrix triipkoodide skaneerimist, kuvab ravimiinfot Eesti ravimiregistri andmete põhjal ning võimaldab hallata koduapteeki, raviskeemi ja hoolealuste profile. 13 planeeritud kasutajalugu 14-st said realiseeritud ning mõlemad testvoorud kinnitasid, et rakendus pakub kasutajatele reaalselt lisaväärtust. Peamiseks kitsaskohaks osutus avaliku triipkoodiandmestiku puudumine ravimiregistri CSV-failis, mistõttu sisestati andmebaasi käsitsi 16 ravimit testandmetega.

Töö näitab, et triipkoodipõhine ravimituvastus on tehniliselt teostatav ning kasutajate poolt positiivselt vastu võetud. Rakenduse täielik potentsiaal realiseerub ravimiregistri tervikliku andmebaasi integreerimisel, mis on ka kõige kriitilisem edasiarenduse suund. Prototüüp loob selleks tugeva tehnilise aluse.

Kasutatud kirjandus

1. Anthropic. (2026). Claude [Tehisintellektiassistent]. <https://claude.ai>
2. Apotheka. (i.a.). Viited ravimiameti ravimiregistrile. <https://www.apotheka.ee/ravimite-register>
3. Capsule. (2026). Capsule: Medication tracker [Mobiilirakendus]. App Store. <https://apps.apple.com/us/app/capsule-medication-tracker/id6744626190>
4. Cohn, M. (2004). User stories applied: For agile software development. Addison-Wesley.
5. EFPIA. (2008). Towards safer medicines supply: A vision for the coding and identification of pharmaceutical products in Europe (Version 4.0). European Commission. https://health.ec.europa.eu/document/download/f1e2e370-d547-4b82-a210-4b8f8642e13e_en
6. ENABLE COST Action. (2024). Tackling medication non-adherence. European Cooperation in Science and Technology. <https://www.cost.eu/tackling-medication-non-adherence/>
7. Figma. (2026). Figma [Disainitööriist]. <https://www.figma.com>
8. Figma. (2026). Figma Make [Tehisintellektipõhine prototüüpimise tööriist]. <https://www.figma.com/make>
9. GitHub. (2026). GitHub Copilot [Programmeerimist toetav generatiivse tehisintellekti tööriist]. <https://github.com/features/copilot>
10. GS1 Eesti. (2015). GS1 süsteemi kasutusjuhend. https://www.gs1.ee/dok/juhendid/GS1_systeemi_kasutusjuhend.pdf
11. Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS Quarterly, 28(1).
12. Lovable. (2026). Lovable [Disainitööriist]. <https://lovable.dev>
13. Medisafe. (2026). Medisafe medication management [Mobiilirakendus]. App Store. <https://apps.apple.com/us/app/medisafe-medication-management/id573916946>
14. Medisafe. (2026). Medisafe medication management [Mobiilirakendus]. Google Play. <https://play.google.com/store/apps/details?id=com.medisafe.android.client>
15. Microsoft. (2025). Build a Blazor movie database app (Part 1 - Create a Blazor Web App). Microsoft Learn. <https://learn.microsoft.com/en-us/aspnet/core/blazor/tutorials/movie-database-app/part-1?view=aspnetcore-8.0>
16. OpenAI. (2026). Codex [Programmeerimist toetav generatiivse tehisintellekti tööriist]. <https://openai.com/codex>
17. Pharmaceutical Society of Ireland. (i.a.). Falsified Medicines Directive. <https://www.psi.ie/practice-supports/falsified-medicines-directive>
18. Ravimiamet. (i.a.). Eesti ravimiregister. <https://www.ravimiregister.ee/>
19. Ravimiamet. (i.a.). Ravimiregistri avalikud andmed [CSV-fail]. <https://www.ravimiregister.ee/?pv=PublicDownloads>
20. Ravimiamet. (2020). Müügilubade taotlemine. <https://www.ravimiamet.ee/ravimid-ja-ohutus/muugiload/muugilubade-taotlemine>

21. SmartPatient GmbH. (2026). MyTherapy: Pill reminder & tracker [Mobiilirakendus]. App Store. <https://apps.apple.com/us/app/pill-reminder-mytherapy/id662170995>
22. SmartPatient GmbH. (2026). MyTherapy: Pill reminder & tracker [Mobiilirakendus]. Google Play. <https://play.google.com/store/apps/details?id=eu.smartpatient.mytherapy>
23. Tallinna Tehnikaülikool. (2023). ISA II - Infosüsteemide arendus II (OIS3_ITB1702). https://ois2.taltech.ee/uusois/aine/OIS3_ITB1702
24. Vabariigi Valitsus. (2016). Ravimiregistri põhimäärus. Riigi Teataja. <https://www.riigiteataja.ee/akt/125052016001>
25. Whobert. (2026). Whobert [Mobiilirakendus]. App Store. <https://apps.apple.com/us/app/whobert/id6762011594>

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Meie, Rando Tiri ja Stella Marie Tohver

- 1 Anname Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) meie loodud teose „Ravimite triipkoodipõhise tuvastamise ja raviskeemide haldamise rakenduse analüüs ja prototüübi arendamine“, mille juhendajateks on Gunnar Piho ja Villem Puusep.
 - 1.1 reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2 üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
- 2 Oleme teadlikud, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autoritele.
- 3 Kinnitame, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

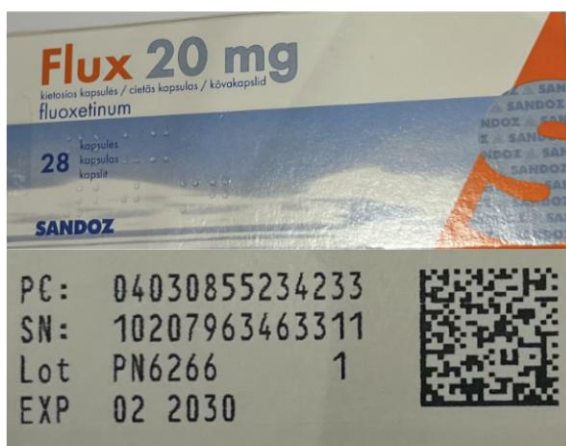
¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või 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 – Andmebaasis olevate ravimite karbid ja triipkoodid

Käesolevas lisis on esitatud rakenduse andmebaasi käsitsi sisestatud 16 ravimi pakendid ja triipkoodid. Joonisel 14 on kujutatud EAN-13 triipkoodiga ravimid ning joonisel 15 DataMatrix triipkoodiga ravimid.



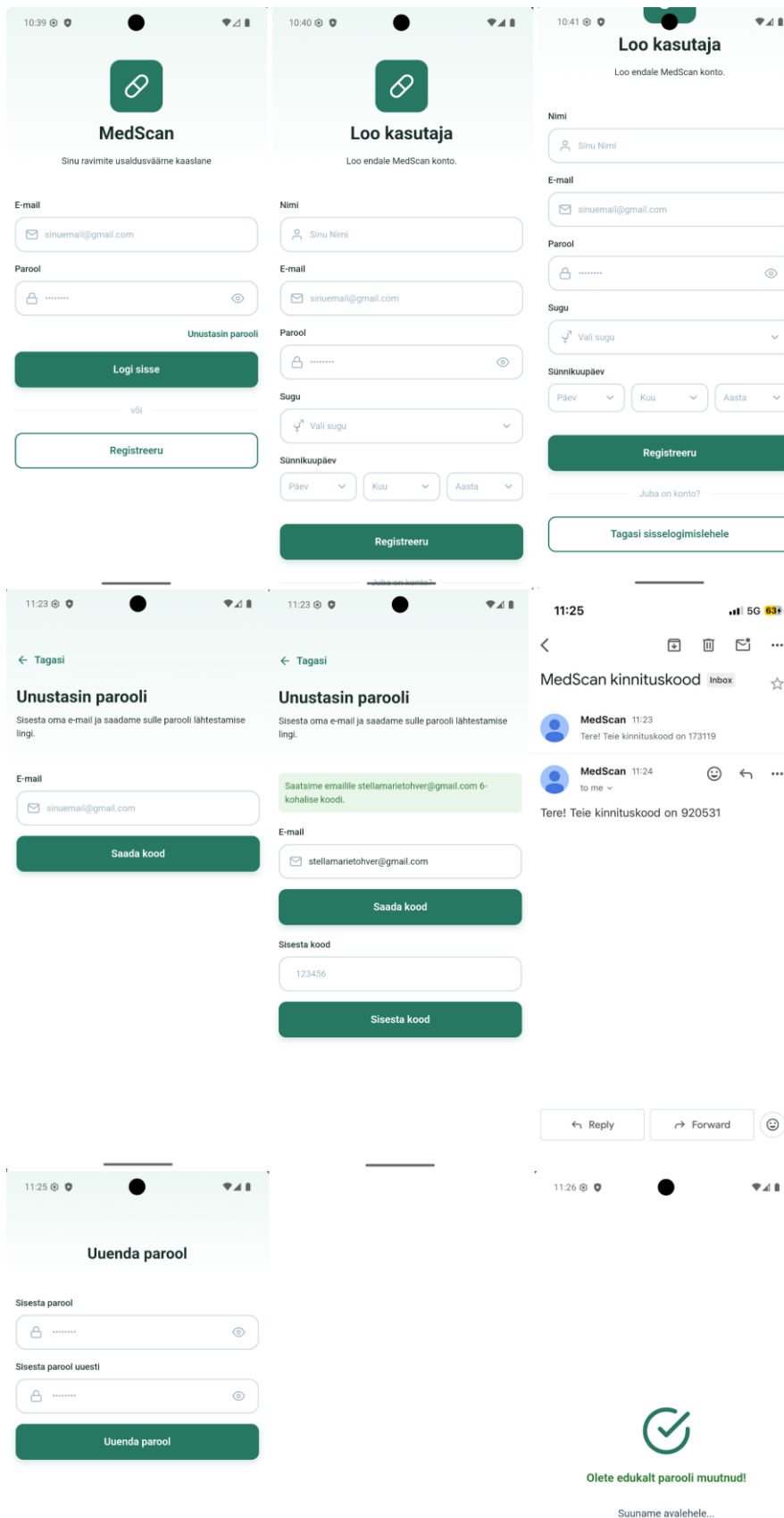
Joonis 10. Andmebaasis olevad EAN-13 triipkoodiga ravimid.



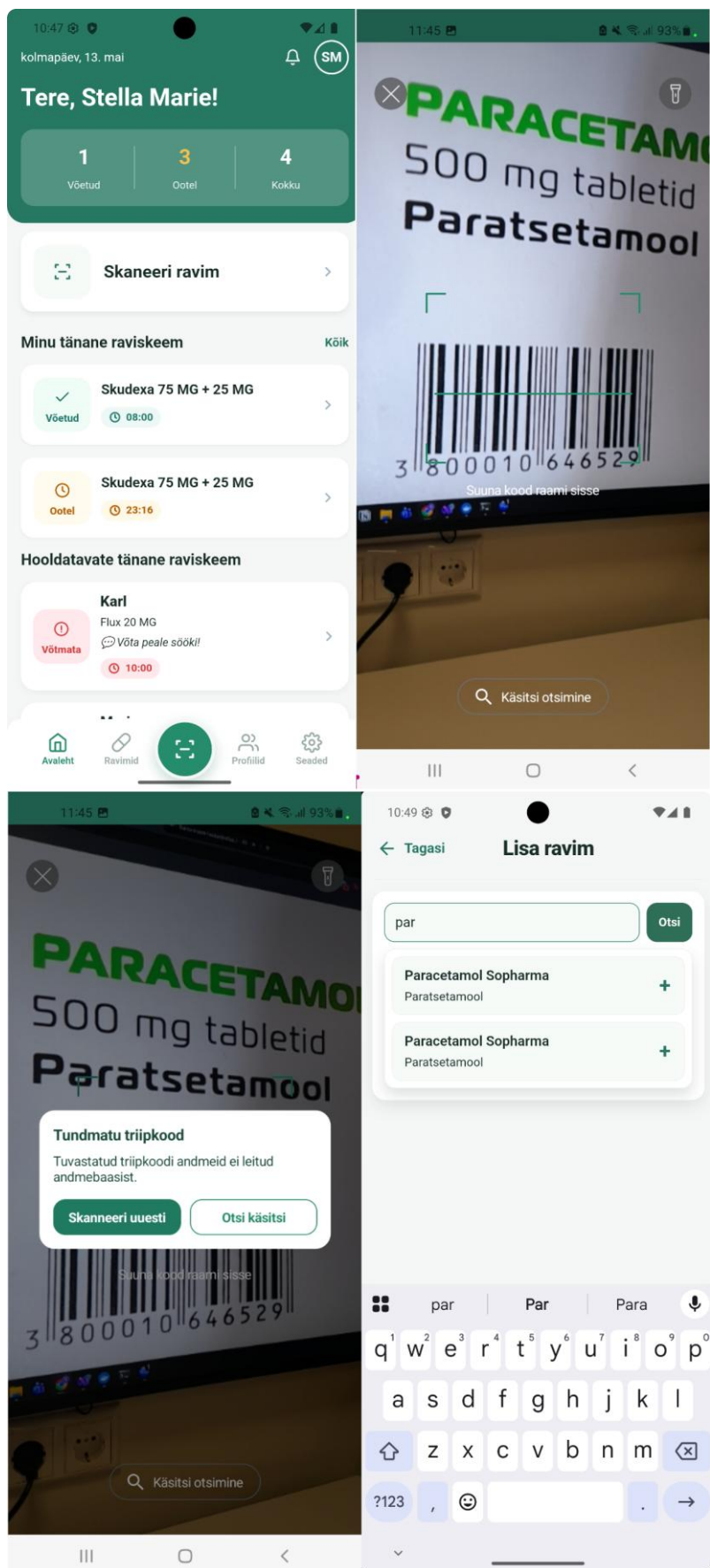
Joonis 11. Andmebaasis olevad DataMatrix triipkoodiga ravimid.

Lisa 3 – Kuvatõmmised valminud prototüübist

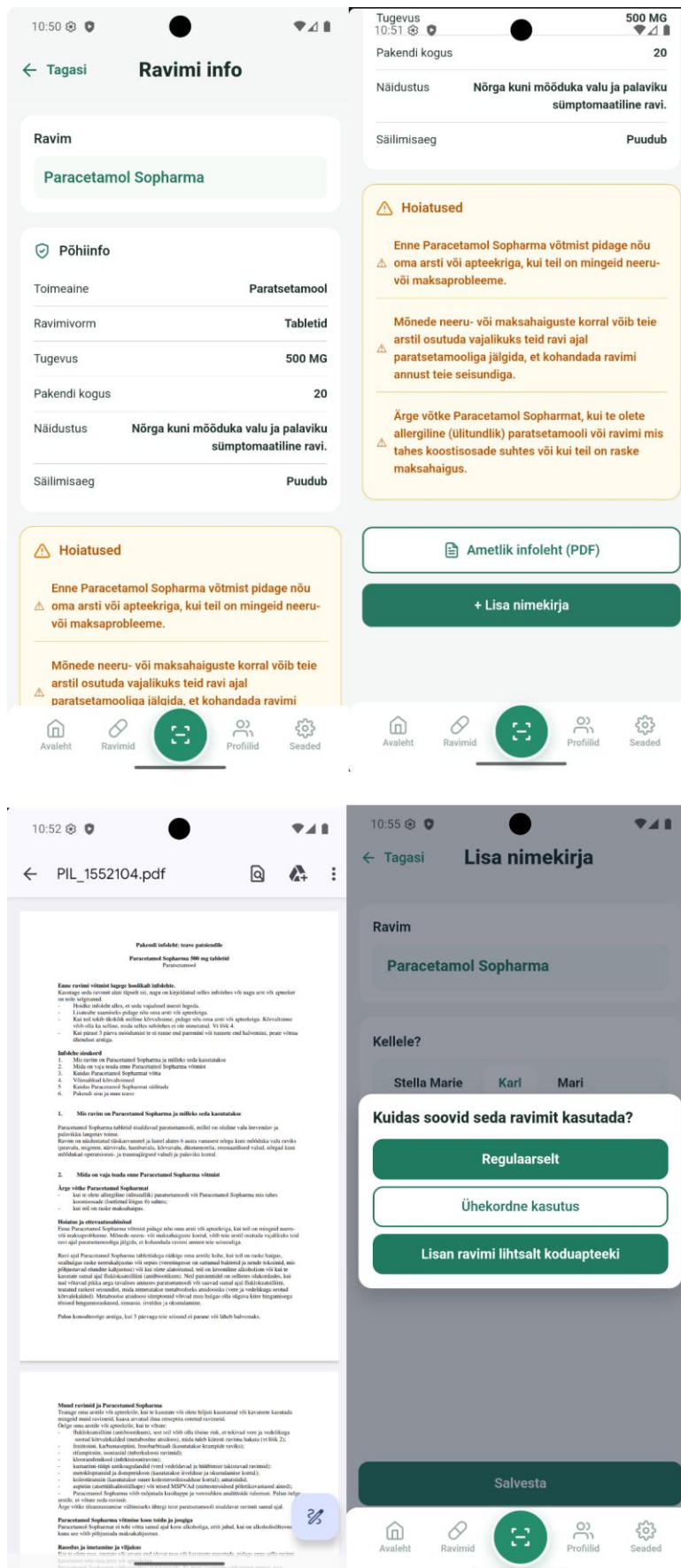
Lisas 3 on esitatud kuvatõmmised MedScan prototüübi realiseeritud funktsionaalsustest. Kuvatõmmised illustreerivad rakenduse kasutajaliidest ja põhifunktsionaalsuste visuaalset lahendust: autentimist, triipkoodi skaneerimist, ravimi infolehte, koduapteegi haldust, raviskeemi ja meeldetuletusi ning hoolealuste haldamist.



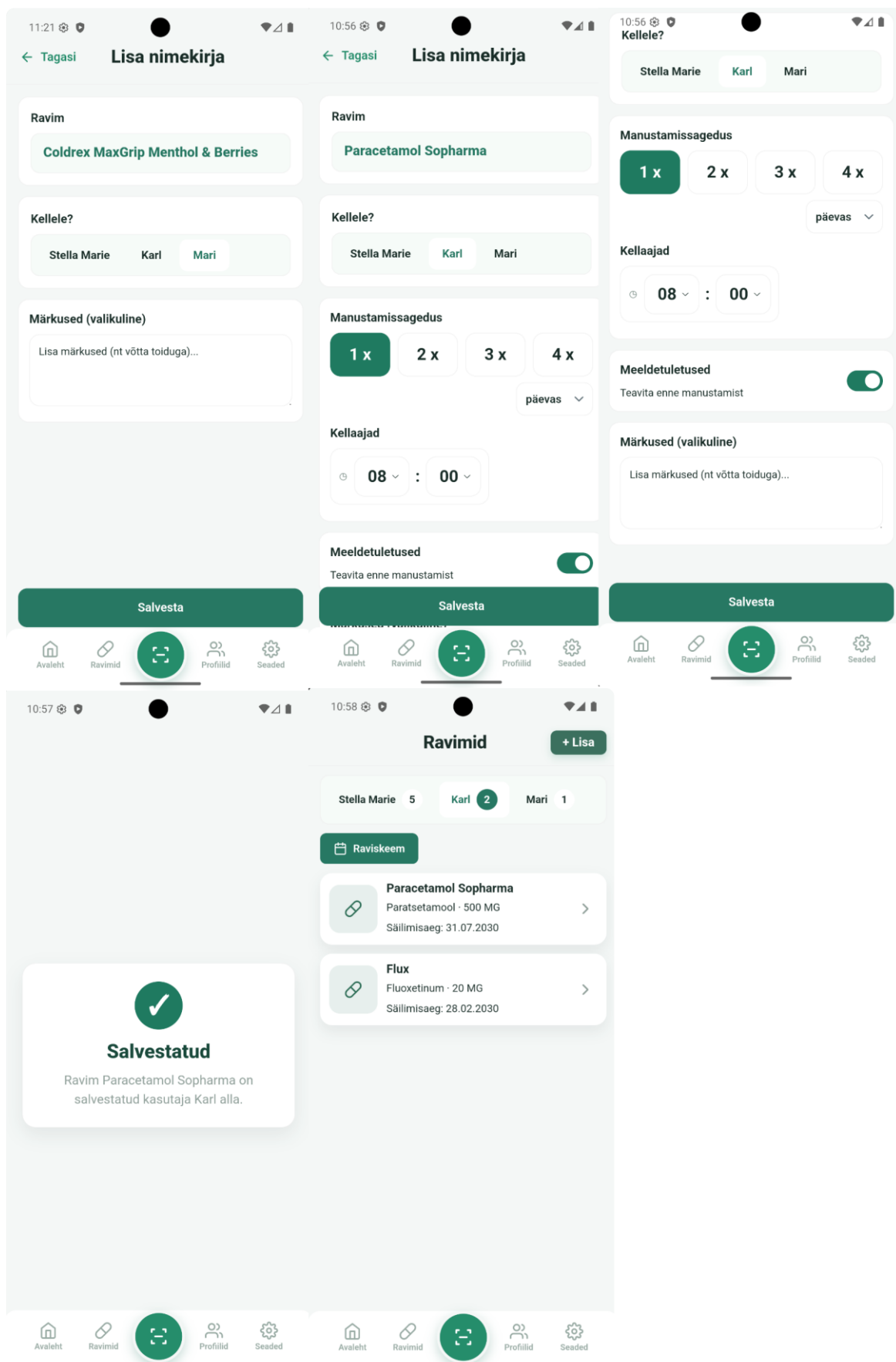
Joonis 12. Kuvatõmmised prototüübi autentimise lehtedest.



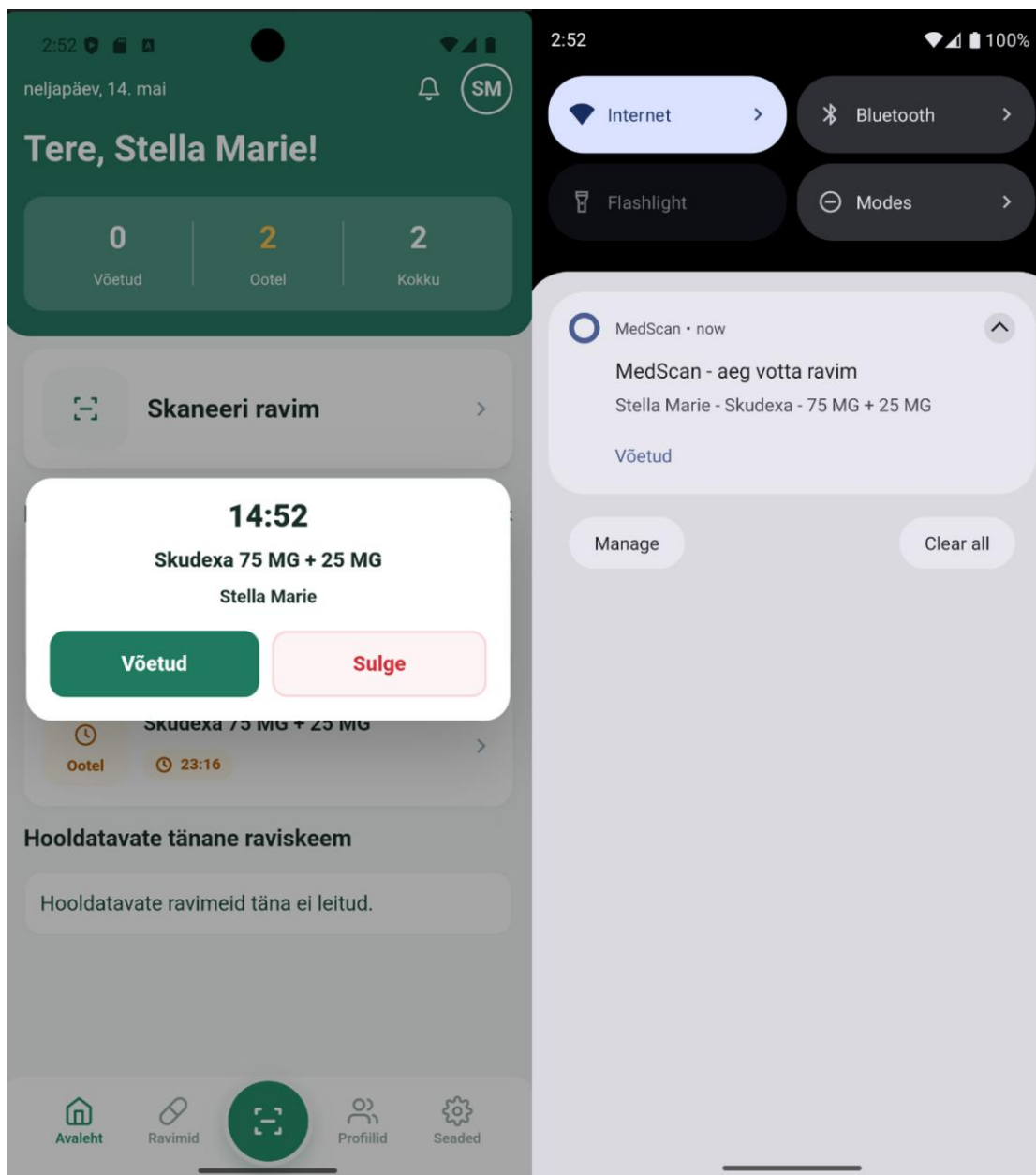
Joonis 13. Kuvatõmmised prototüübi skaneerimise lehtedest.



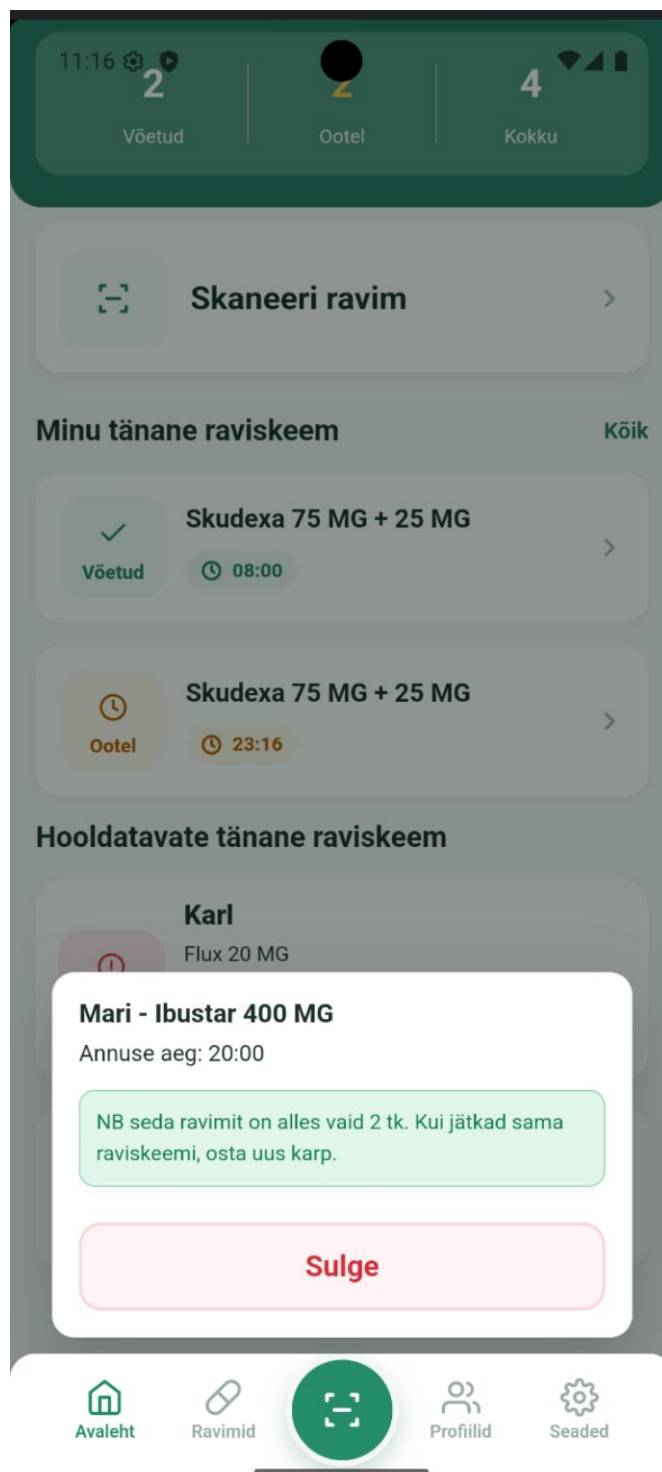
Joonis 14. Kuvatõmmised prototüübi ravimi infolehest.



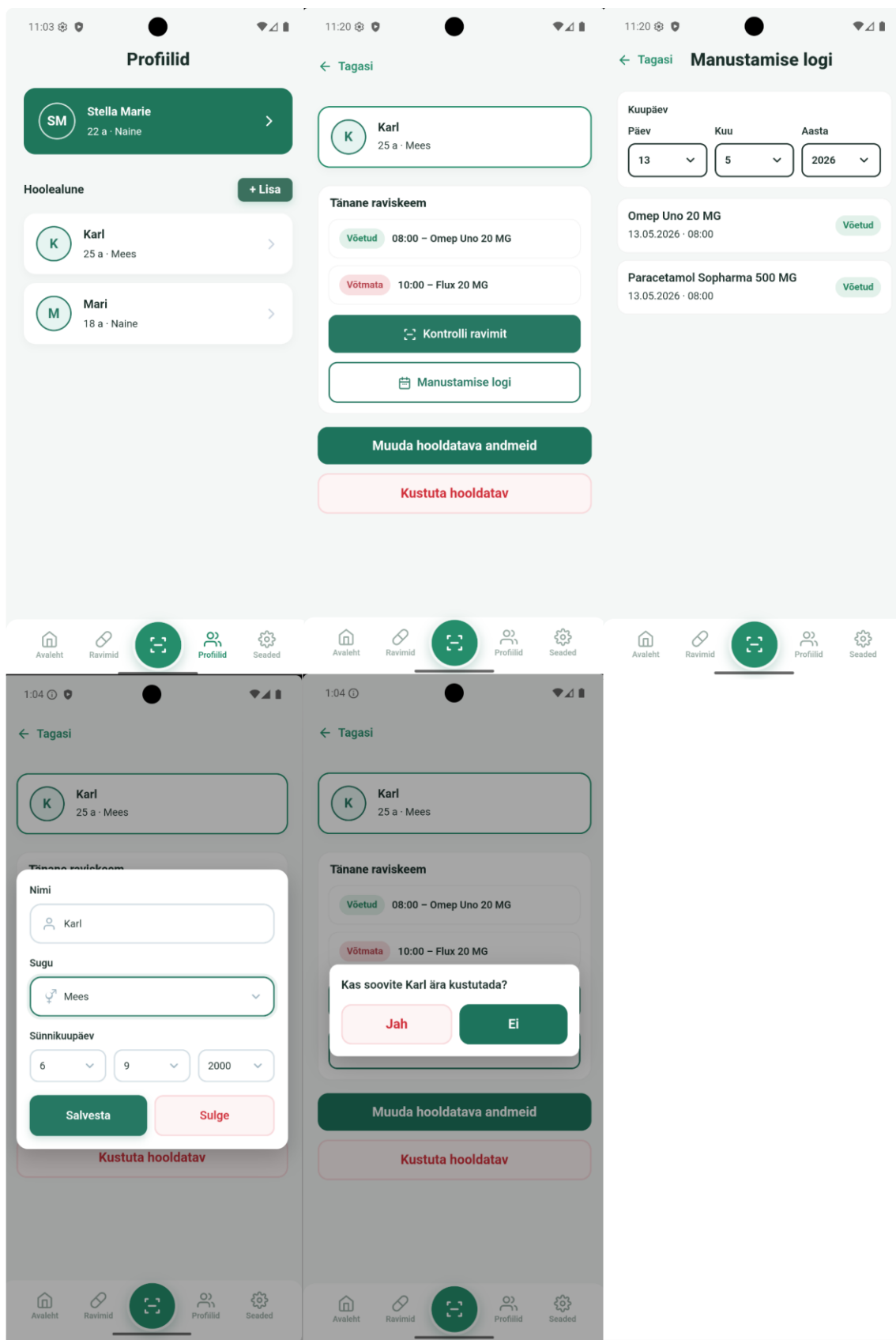
Joonis 15. Kuvatõmmised prototüübi lisa nimekirja ja ravimid lehtedest.



Joonis 16. Kuvatõmmised prototüübi meeldetuletustest.



Joonis 17. Kuvatõmmis prototüübi madala koguse teavitusest.



Joonis 18. Kuvatõmmised prototüübi hoolealuste haldamise lehest.