TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Ron-Aran Paju

Amphora Infohaldus OÜ rakenduse ja kaheastmelise autentimise arendamine

Bakalaureusetöö (9 EAP)

Juhendaja: Kristiina Rahkema, PhD

Tartu 2025

Amphora Infohaldus OÜ rakenduse ja kaheastmelise autentimise arendamine

Lühikokkuvõte:

Käesolevas bakalaureusetöös arendati mobiilirakendus ja QR-koodil põhinev kaheastmeline autentimislahendus ettevõttele Amphora Infohaldus OÜ. Töö eesmärk oli luua lahendus, mis võimaldab kasutajatel turvaliselt siseneda Amphora mobiilirakendusse kaudu ning näha seal tööülesandeid. Töö käigus analüüsiti Eestis kasutusel olevaid autentimisviise, sh Mobiil-ID, Smart-ID ning QR-koodil põhinev autentimine, ning testiti lahendusi nagu LHV mobiilirakendus, Eesti terviseportaal ja WhatsApp. Töö tulemusena valmis rakendus, mis täitis enamiku funktsionaalseid ja mittefunktsionaalseid nõudeid. Rakendus võimaldab QR-koodiga sisselogimist, tööülesannete kuvamist ja serveriga turvalist andmevahetust.

Võtmesõnad: kaheastmeline autentimine, QR-kood, mobiilirakendus, .NET, REST API

CERCS: P175 Informaatika, süsteemiteooria

Development of the Amphora Infohaldus OÜ Mobile Application and Two-Factor Authentication

Abstract:

This bachelor’s thesis focuses on developing a mobile application and a two-factor authentication solution for Amphora Infohaldus OÜ. The goal was to create a secure way for users to log in to the Amphora information system via a mobile app and view assigned tasks. The work included an analysis of common authentication methods used in Estonia, such as Smart-ID, Mobile-ID, and QR-code-based login. Practical testing was performed using applications like LHV Bank, the Estonian health portal, and WhatsApp. As a result, a functional prototype was created that meets most of the functional and non-functional requirements.

Keywords: two-factor authentication, QR code, mobile application, .NET, REST API

CERCS: P175 Informatics, systems theory

Sisukord

1. Sissejuhatus

Tänapäeva kiiresti arenevas digitaalses keskkonnas on andmete ja kasutajakontode turvalisus muutunud kriitilise tähtsusega osaks tarkvara arendusest. Küberruumis toime pandavate kuritegude arv on kiires kasvutrendis. Näiteks registreeriti 2024. aastal Eestis 6515 mõjuga intsidenti, mis on keskmiselt ligikaudu 18 intsidenti päevas. 4224 juhtumit sellest olid seotud õngitsus rünnakutega. Võrreldes 2023. aasta intsidentide arvuga 3314 oli 2024. aasta juhtumite arv peaaegu kahekordistunud [1]. Tavapärase kasutajanime ja parooli kombinatsiooni kasutamine ei paku enam piisavat turvataset, kuna jõurünnakute ja pahavara levik on kasvatanud riske andmete kuritarvitamiseks. Selle probleemi lahendusena on järjest enam kasutusele võetud mitmeastmelist autentimist, mis nõuab kasutaja tuvastamiseks mitme sõltumatu teguri kombinatsiooni, mis võib päästa andmed ka rünnaku ajal.

Amphora Professional on Eestis kasutatav infohaldussüsteem, mis võimaldab omavalitsustel ja organisatsioonidel mugavalt hallata dokumente ning töövooge ühest keskkonnast. Varasemalt on Amphora kasutajaliides olnud vaid veebipõhine, mis ei paku piisavalt paindlikkust mobiilseks kasutamiseks tänapäevaste standardite järgi. Selleks, et veenduda rakenduse turvalisuses on vajalik pakkuda süsteemile täiendavat autentimise kihti, et kõrvalised isikud ei pääseks delikaatsetele andmetele ligi. Eestis on kasutusel erinevaid kaheastmelise autentimise võimalusi, nt Smart-ID, Mobiil-ID ja QR-koodil põhinev kaheastmeline autentimine. Viimane on end praktikas tõestanud kui kiire ja mugav lahendus, mis ei nõua lisalepinguid. Seda on edukalt implementeeritud paljudes populaarsetes teenustes, nagu näiteks populaarne suhtlusrakendus WhatsApp. Rakendades sama lahendust Amphora süsteemile, on võimalik vähendada pahaloomulist konto ülevõtmise tõenäosust.

Käesolev bakalaureusetöö on jaotatud kuueks peatükiks. Antud peatükis ehk esimeses on käsitletud töösse sissejuhatus. Teises peatükis antakse ülevaade tööga seonduvast taustast ning kirjeldatakse Amphora infohaldussüsteemi ja tehnoloogiaid, millel arendatav rakendus põhineb. Kolmandas peatükis on kirjeldatud kasutatud metoodikat, sealhulgas nõudeid ja arendusprotsessi. Neljandas peatükis käsitletakse saavutatud tulemusi, analüüsitud rakenduste kohta, loodud rakenduse struktuuri ja funktsionaalsuse toimimist ning samuti ka autentimise lahendusest. Viiendas peatükis arutletakse, kuidas edenes rakenduse arendus, kui palju soovitud nõuetest sai

täidetud, tehakse soovitusi ja ettepanekuid edasisteks arendusteks ning käsitletakse saadud tagasisidet arendatud rakednduse osas. Viimases peatükis on kokkuvõte antud bakalaureusetööst.

2. Taust

2.1 Autentimine

Autentimine on toiming, mis kinnitab kasutaja identiteedi. Üks levinud viis selleks on kasutajanime ja parooli kontrollimine [2]. Tänapäeval on kiire tehnoloogia arenguga järjest enam esile kerkinud vajadus tugevamate autentimismeetodite järele, kuna pidevalt areneb pahavara ja kasvab petiste arv, kelle eesmärgiks on andmete kuritarvitamine.

2.1.1 Mitmeastmeline autentimine

Tavapärase üheastmelise autentimise asemel on saanud viimaste aastatega levinuks mitmeastmeline autentimine. Mitmeastmelise autentimise puhul on kasutaja kinnitamiseks vajalik tõestada identiteeti mitmel sõltumatul viisil [3]. Näiteks veebirakenduste puhul võib peale kasutajanime ja parooli sisestamist saata rakendus kinnituskoodi SMS-sõnumina määratud telefoninumbrile või kasutada koodi edastamiseks mõnda kolmanda osapoole rakendust.

Kaheastmelist autentimist kasutatakse põhiliselt keskkondades, kus on tegemist tehingutega näiteks e-poed, pangad, hasartmängud. Lisaks ka teenused, kus töödeldakse tundlikke isikuandmeid nagu näiteks erinevad sotsiaalmeedia portaalid, kodanikuportaal, terviseandmetele ligipääs jne [4]. See tagab turvalisuse juhuks, kui parool ja kasutajanimi on kolmanda osapoole poolt varastatud või jõuründe (ingl *brute force attack*) abil tuvastatud - kasutaja saab veel võimaluse sisselogimist kinnitada või katkestada.

2.1.2 QR-koodiga autentimine

Lisaks eelnevalt mainitud viisidele on levinud ka QR-koodiga autentimine. Meetod toimib üldjuhul nii, et kasutaja peab olema eelnevalt veebi- või mobiilirakendusse sisse logitud ning seejärel skaneerib oma nutiseadmega veebilehel või rakenduses kuvatava unikaalse QR-koodi. Peale skaneerimist tuvastatakse kasutaja automaatselt ja autentitakse keskkonda. Samas võib QR-koodi autentimisel olla siiski märkimisväärne turvarisk, kui pahatahtlikud osapooled loovad kuritahtlikke QR-koode või asendavad õigeid koode enda loodutega [5].

2.1.3 Autentimislahendused Eestis

Eestis on paljudes populaarsetes teenustes ja rakendustes kasutusel Mobiil-ID ja Smart-ID, mis töötavad samuti kaheastmelise autentimise põhimõttel. Lisaks on levinud SMS-põhine autentimine, kus kasutaja siseneb keskkonda esmalt kasutajanime ja parooliga ning seejärel kinnitab seansi SMS-iga saadud koodiga [6].

Smart-ID on autentimiseks ja allkirjastamiseks mõeldud rakendus, mis võimaldab kasutajal siseneda e-teenustesse ja lisaks pakub ka allkirjastamise võimalust [7]. Mobiil-ID on SIM-kaardil põhinev digitaalse identiteedi lahendus, millega on võimalik kõrge turvalisusega autentida end e-teenustes ning anda digitaalallkirju otse mobiiltelefonist [8].

Eesti kontekstis on oluline mainida ka Riigi Autentimisteenust (GovSSO), mis on 2022. aastal käivitatud autentimislahendus avaliku sektori e-teenustele. Lahendus toetab kõiki Eestis levinud autentimisvahendeid, sealhulgas ID-kaarti, Mobiil-ID-d, Smart-ID-d. GovSSO peamine eelis seisneb selles, et kasutajal on võimalik autentida ühe korra ning liikuda erinevate e-teenuste vahel ilma uuesti sisse logimata. Süsteem on üles ehitatud vastavalt OIDC (ingl OpenID Connect) standardile, mis tagab kõrge turvalisuse taseme ning ühilduvuse rahvusvaheliste autentimise standarditega [9].

Eestis pakutavatest autentimise lahendustest on Smart-ID üks lihtsamaid ja populaarsemaid, sest kasutajale ehk eraisikutele on selle kasutamine tasuta. Äriklientidele kehtivad hinnad sõltuvalt kasutajate arvust ja vajadustest - näiteks 2500 sessiooni puhul on miinimumhind 220 € kuus, samas kui 100 000 sessiooni puhul on tasu vähemalt 3960 € kuus [10]. Mobiil-ID teenuse kuutasu on tarbijatele ligikaudu 1 euro kuus, olenevalt mobiilioperaatorist ning on teenusepakkujale sarnase hinnastusega nagu Smart-ID [11].

Lisaks eelnevalt mainitud autentimise viisidele on veel lahendusi. Näiteks kasutab Google autentimise viisi, kus parooli edukal sisestamisel saadab teenusepakkuja kasutaja poolt seatud seadmele teate veenmaks, kas sisselogimine lubada või mitte. Teine Google poolt pakutud võimalus on Google Authenticator rakenduse kasutamine, mis eeldab usaldatud seadmesse Google poolt loodud autenimiseks mõeldud rakenduse installeerimist. Soovitud keskkonda logitakse sisse

kasutades näiteks kasutajanime ja parooli ning seejärel palub autentimine, sisestada koodi, mis Google Authenticator rakenduses on kuvatud, koodi uuendatakse iga 30 sekundi järel [12].

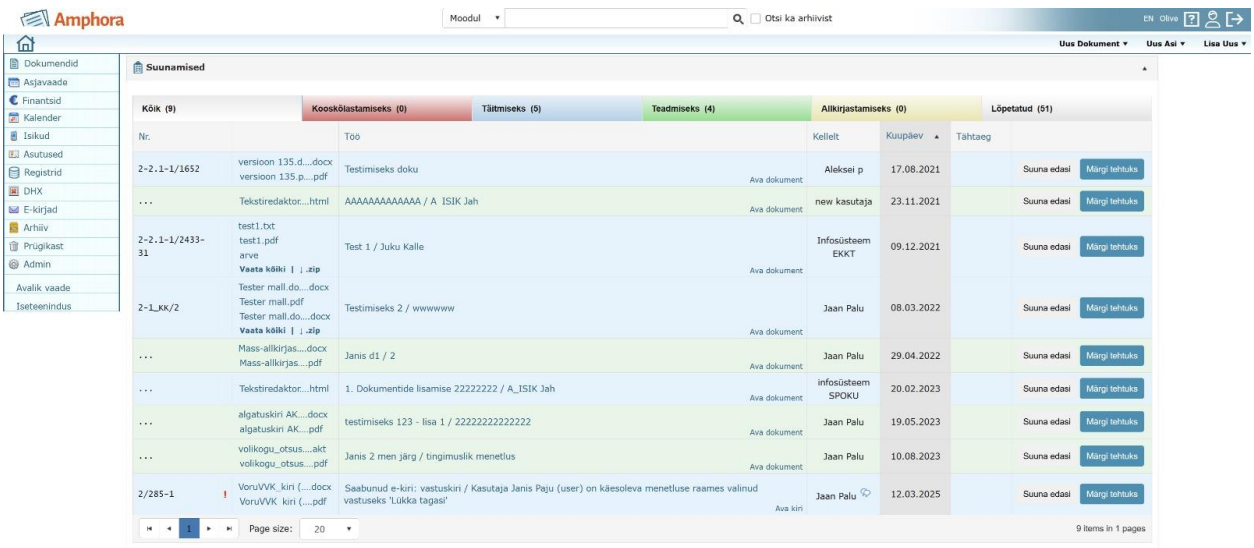
2.2 Amphora

Amphora Infohaldus OÜ on spetsialiseerunud omavalitsuste infohaldusele teenusega Amphora Professional, mis võimaldab hallata suurel hulgal dokumente lihtsalt ja struktureeritult. C# keeles kirjutatud süsteem hõlmab kõiki olulisi dokumendihalduse etappe – alates loomise ja registreerimise kuni menetlemise ning digiallkirjastamiseni. Näiteks mitmekümne dokumendi allkirjastamine on võimalik vaid mõne sammuga, säästes aega kasutajatele, kellel on töödelda hulk dokumente päevas. Dokumente on võimalik lisada arvutist, koostada otse rakenduses, skaneerida või edastada otse Microsoft Wordi abil. Iga toiming on jälgitav, tagades läbipaistvuse ja turvalisuse. Samuti võimaldab täpne otsing leida kiiresti infot nii dokumentide metaandmetest kui ka sisust, sealhulgas erinevatest failivormingutest nagu Word, Excel ja PDF. Joonisel 1 on nähtav kuvatõmmis Amphora Professional kasutajaliidest menetlusülesannete haldamisel, kus vasakul oleval menüüs on nähtaval ka muud võimalikud teenused ja funktsionaalsused.

Kirjavahetuse funktsioon lihtsustab nii saabuvate kui ka väljaminevate dokumentide registreerimist ja jälgimist. Kirjade koostamine toimub eelmääratud mallide põhjal ning allkirjastamine otse süsteemis. Sõnumite edastamiseks saab kasutada nii e-kirju kui ka DHX-liidest ehk Dokumendivahetuskiht, mis tagab andmevahetuse erinevate asutuste vahel. Amphora võimaldab ka koondada ja hallata omavahel seotud dokumente, kasutades virtuaaltoimikuid ja automaatseid töövooge, mis aitavad suunata dokumente õigetele isikutele ning menetleda neid kas individuaalselt või tervikuna. Amphora süsteemis on ka Microsoft Outlooki tugi, mis võimaldab nii saabunud kui ka saadetud e-kirju registreerida Amphorasse. Samuti toetab süsteem kalendri funktsiooni ühise ajaplaneerimise jaoks, võimaldades vaadata ja hallata enda ning kolleegide sündmusi. Lisaks sisaldab Amphora funktsioone arutelude pidamiseks, kontaktide haldamiseks ning dokumentidega seotud info grupeerimiseks. E-kirjade integreerimine võimaldab saabunud dokumentide automaatset registreerimist ning nende edastamist õigetele osapooltele.

Süsteem on täielikult kohandatav, pakkudes administraatoritele võimalusi kasutajaõiguste määramiseks, töövoogude seadistamiseks ja dokumentide vormide loomiseks. Avaliku teabe seadusest tulenevaid nõudeid silmas pidades on loodud ka avalik vaade, mis võimaldab

juurdepääsu dokumentidele [13]. Firmal on plaan tulevikus laiendada Amphora Professionali kättesaadavust mobiilseadmetele, et kasutajad saaksid infohaldussüsteemi funktsionaalsusele ligi pääseda ka väljaspool kontorit. Selle eesmärgi saavutamiseks on käesolevas töös arendatav mobiilirakendus, mis võimaldab kasutajatel turvaliselt autentida ning näha oma tööülesandeid.



Joonis 1. Amphora professional avavaade.

2.3 Kasutatud tehnoloogiad

Amphora mobiilirakenduse loomisel on kasutatud mitmeid tehnoloogiaid. Järgnevates alampeatükkides antakse ülevaade valitud arendustehnoloogiatest, kirjeldades nende põhilisi omadusi. Tutvustatakse kasutatud programmeerimiskeelt ja raamistikku, andmevahetuse protokolle, andmebaasisüsteemi ning rakenduse arhitektuurilist mustrit.

2.3.1 C# ja .NET MAUI raamistik

C# (ingl C Sharp) on Microsofti poolt loodud objektorienteeritud ja tüübiturvaline programmeerimiskeel, mis avaldati esmakordselt 2002. aastal koos .NET raamistikuga [14]. Aasta 2024 seisuga on C# kujunenud maailma kaheksandaks populaarseimaks programmeerimiskeeleks tarkvaraarenduses [15]. C# pakub kaasaegseid programmeerimisvõimalusi, sealhulgas

asünkroonset programmeerimist, automaatset mäluhaldust ja tugevat tüübikontrolli, mis võimaldavad luua hästi skaleeruvaid ja töökindlaid lahendusi [14].

.NET MAUI (ingl Multi-platform App UI) on Microsofti 2022. aastal ilmunud raamistik, mis võimaldab ühe projekti alusel arendada rakendusi mitmele operatsioonisüsteemile, sealhulgas Windowsile, macOS-ile, Androidile ja iOS-ile. Tegemist on Xamarin Formsi edasiarendusega, mis võrreldes eelmise versiooniga pakub paremat jõudlust, lihtsamat arhitektuuri ja rohkem struktureeritud arenduskogemust. .NET MAUI koondab platvormispetsiifilised API-d ühe kihina, vähendades vajadust luua eraldi lahendusi iga platvormi jaoks [16]. Rakenduste kasutajaliideste loomiseks toetab .NET MAUI XAML-i (ingl Extensible Application Markup Language) lahendust, kuid võimaldab arendada liideseid ka täielikult C# keeles. Oluliseks eeliseks on ka sisseehitatud tugi Blazori ja MAUI Hybrid-rakendustele, mis võimaldab veebitehnoloogiate integreerimist mobiili- ja töölauarakendustesse [16,17].

2.3.2 REST API

REST rakendusliides (ingl REST API) on lahendus, mis võimaldab rakendustel ja serveripoolsetel teenustel omavahel suhelda ühtsete ja standardiseeritud HTTP-päringute kaudu [18]. HTTP (ingl HyperText Transfer Protocol) on kliendi-serveri protokoll, kus klient (tavaliselt veebibrauser) saadab serverile HTTP-päringu, millele server vastab vastava vastusega, mis võib olla JSON (ingl JavaScript Object Notation), HTML, XLT, Python, PHP või tavaline tekst. [18,19]. REST API toetab meetodeid nagu nagu GET, POST, PUT ja DELETE, mis määratlevad andmete lugemise, loomise, muutmise ja kustutamise operatsioone. Lisaks võimaldab REST API arhitektuur paindlikkust ja lihtsat integreerimist erinevate platvormide vahel [18].

Süsteemi autentimise võimaluste loomiseks on REST API sobiv, sest see võimaldab kindlat viisi päringute teostamisel. See muudab kõik päringud ühtseks ning lihtsustab dokumenteerimist. Autentimismehhanismi kaudu saab Amphora platvorm ja mobiilirakendus turvaliselt kasutajate andmeid vahetada. REST API standardite järgimine tagab ka ühilduvuse rahvusvaheliste autentimisprotokollidega, muutes süsteemi laiendatavaks ja tulevikukindlaks [18].

2.3.3 QR-KOOD

QR-kood (ingl *Quick Response Code*) on kahemõõtmeline vöötkood (Joonis 1a), mis on loodud teabe kiireks ja võimalikult lihtsaks edastamiseks, võimaldades seadmetel, nagu nutitelefonidel või tahvelarvutitel, skaneerida ning tõlgendada visuaalsel kujul esitatud andmeid. QR-koodi eelkäijaks oli ühemõõtmeline vöötkood, mida kasutatakse laialdaselt näiteks jaekaubanduses toodete märgistamiseks (Joonis 1a), kuid QR-kood (Joonis 1b) on suuteline salvestama märksa rohkem teavet tänu oma kahemõõtmelisele struktuurile. QR-koodid koosnevad mustadest ja valgetest ruutudest ruudustikus, mis kodeerivad andmeid – näiteks veebiaadresse, kontaktandmeid, makseinfot või autentimise pöördumisi [5].

Joonis 2. Erinevad vöötkoodide tüübid: (a) ühemõõtmeline vöötkood ja (b) QR-kood.

Tänapäeval on QR-koodide rakendusalad väga laiad – alates turundusest ja piletimüügist kuni rakendustes autentimiseni [20]. Turvalisuse vaatenurgast on QR-koodil nii eeliseid kui ka riske. Positiivseks on kontaktivabad lahendused, mis võimaldavad kasutajatel autentida end kiiresti ja mugavalt skaneerimise abil, ilma käsitsi parooli sisestamata. Näiteks kasutatakse QR-koodi abil autentimist mobiili- või veebirakenduses, kus kasutaja skaneerib oma seadmega teenuse poolt genereeritud unikaalse QR-koodi ning saab sellega automaatselt sisse logida [21]. Sellist autentimist toetavad paljud tuntud rakendused ja platvormid, nagu näiteks WhatsApp Web, Google Authenticator ja Microsoft Authenticator, mis võimaldavad turvalist ligipääsu kontole mitmeastmelise autentimise osana [22]. QR-koodi kasutamisel autentimisel on oluline, et kasutaja oleks eelnevalt seotud seadmega näiteks mobiilirakenduse või veebirakenduse kaudu, et

autentimise käigus saaks kinnitada kasutaja isiku. QR-kood võib sisaldada krüpteeritud sessiooniteavet, mida server suudab pärast skaneerimist valideerida.

Siiski on QR-koodidega seotud ka turvariske. Levinud oht on pahatahtlike QR-koodide kasutamine, kus kood suunab kasutaja kahjulikele veebilehtedele, varastab andmeid või üritab installida pahavara seadmesse. Selliste rünnakute vältimiseks tuleb veenduda, et QR-kood pärineb usaldusväärsest allikast ja kasutatav rakendus toetab turvalist dekodeerimist ja kontrolli [23]. Eestis on QR-koodide kasutamine kasvanud koos e-teenuste arenguga, kus koodi kasutatakse näiteks elektrooniliste piletite valideerimiseks või ligipääsuks lukustatud infosüsteemidesse.

2.3.4 MSSQL

Microsoft SQL Server (ingl MSSQL) on Microsofti poolt arendatud relatsiooniline andmebaasihaldussüsteem (ingl RDBMS – Relational Database Management System). Tänu regulaarsetele uuendustele ja laiadasele kasutusele on MSSQL jätkuvalt üks populaarsemaid andmebaasiplatvorme nii pilve- kui ka lokaalseks kasutuseks [24]. Antud töös on autentimiseks kasutusel üks tabel nimega sys_users_device, kus on välja toodud väljad kasutaja, seadme ja tokeni jaoks, tabel on nähtav jooniselt 3.

Home\MSSQL2022.a....sys_users_device* | Object Explorer Details

Column Name	Data Type	Allow Nulls
id	int	☐
sys_users_id	int	☐
token	uniqueidentifier	☐
token_expires	datetime	☐
device_id	nvarchar(255)	☑
device_name	nvarchar(500)	☑
device_descr	nvarchar(4000)	☑
activated	datetime	☑
create_time	datetime	☐
		☐

Joonis 3. MSSQL andmetabel autentimiseks

2.3.5 Model-View-ViewModel

MVVM (ingl *Model-View-ViewModel*) kujutab endast tarkvaraarhitektuuri mustrit, mis struktureerib rakenduse kolme põhikomponendina: kasutajaliides (ingl *View*), andmemudel (ingl *Model*) ning neid ühendav vahekiht (ingl *ViewModel*). Selline arhitektuuriline lähenemine võimaldab säilitada koodi struktureeritud ülesehituse, tõhustada komponentide testitavust ning lihtsustada rakenduse edasist hooldust ja arendamist [25].

3. Meetod

Antud Bakalaureusetöö eesmärk on välja töötada mobiilirakendus, mis võimaldab turvaliselt ja mugavalt autentimist QR-koodi abil mobiilirakendusse ning näha selle kaudu tööülesandeid kasutaja vaatest. Samuti uuritakse kuidas Eestis kasutusel olevates populaarsetes rakendustes on lahendatud autentimine. Enne käesoleva töö alustamist pakkus Amphora Infohaldus OÜ klientidele juurdepääsu ainult veebirakenduse kaudu, mis toetas mitmeid autentimismeetodeid: tavapärast kasutajanime ja parooli viisi kui ka täiendavaid kaheastmelisi autentimisvõimalusi. Lisaturvalisuse tagamiseks on võimalik rakendada ka kasutajakontoga seotud IP-piiranguid, mis võimaldavad iga kasutajakonto ligipääsu täpselt reguleerida. Funktsioon on eriti väärtuslik väliste süsteemide liidestamisel, mis kasutavad veebiteenuseid, kuna rakendusserveri tasemel saab igale integreeritud süsteemile seada spetsiifilised IP-piirangud. Üheastmelise autentimise jaoks on kasutusel jõurünnaku vastane kaitse viie ebaõnnestunud sisselogimiskatse piirang. Enne mobiilirakenduse arendust uuritakse Eestis kasutusel olevaid mitmetasandilisi autentimislahendusi, saadud tulemusi rakendatakse mobiilirakenduse arendamisel.

3.1 Mitmetasandilise autentimise implementeerimine rakendusele

Käesoleva bakalaureusetöö esimese osa eesmärgiks on analüüsida Eestis kasutusel olevaid mobiilirakenduste autentimislahendusi, keskendudes mitmetasandilistele autentimismeetoditele. Analüüsi eesmärk on tuvastada praktikaid ja lahendusi, mida saaks rakendada samuti Amphora mobiilirakenduse arendamisel, et tõsta kasutajakogemuse mugavust ja turvalisust.

3.1.1 Analüüsi planeerimine

Analüüsi läbiviimiseks on plaanis erinevate rakenduste autentimise viiside uurimine. Analüüsitavate rakenduste valimiseks keskenduti põhiliselt igapäevases kasutuses olevatele rakendustele ning riiklikele e-teenustele, mis töötlevad delikaatseid andmeid ning võimaldavad erinevaid autentimise viise. Rakendusi valides arvestatakse erinevate autentimismeetodite olemasolu ning rakenduse kättesaadavust testimiseks. Nende kriteeriumite alusel tagatakse, et analüüs hõlmab asjakohaseid ja kohalikul turul pakutavaid lahendusi.

3.1.2 Planeeritav rakenduste valim ja testimine

Testimiseks on planeeritud valida kolm erinevat rakendust, mis esindavad erinevaid valdkondi, autentimislahendusi ning erinevaid eesmärke. Esimesena on kavas analüüsida LHV panga mobiilirakendust, kus on kajastatud mitmeid autentimismeetodeid. Teiseks uuritakse Eesti terviseportaali veebirakenduse lahendust, mis rakendab rangeid turvanõudeid koos autentimisega tundlike terviseandmete kaitseks. Kolmandana võetakse vaatluse alla WhatsApp mobiilirakendus, mis kasutab QR-koodipõhist autentimist veebiversiooni ja mobiilirakenduse sidumiseks. Iga rakenduse testimisel järgitakse protseduuri, alustades rakenduse installeerimisest ning jätkates sisselogimise protsessi alustamisega ja tööga kattuvate pakutavate autentimisvalikute dokumenteerimisega - testimisel ei testita PIN-kalkulaatorit ja töös mitte kasutatavaid lahendusi. Seejärel läbitakse erinevad autentimismeetodid algusest lõpuni, katkestatakse autentimisprotsessi erinevates etappides veakäsitluse hindamiseks ning testitakse turvaprotseduure teadlikult valede sisendite sisestamisega. Lisaks hinnatakse lahenduse töökindlust ning analüüsitakse autentimisprotsessi üldist kasutajasõbralikkust.

3.2 Funktsionaalsed ja mittefunktsionaalsed nõuded rakendusele

Funktsionaalsed nõuded on ette seatud ootused programmile kasutuse ja toimise poolelt ning need mõjutavad kasutaja kogemust rakendust kasutades [26]. Enne arenduse algust seati Amphora poolt paika funktsionaalsed nõuded rakendusele, mis valmis rakendusel peavad olema. Mittefunktsionaalsed nõuded viitavad aspektidele, mida kasutaja otseselt kasutada ei saa, näiteks nuppu või ikooni [26]. Järgmised nõuded on Amphora poolt seatud rakenduse jaoks.

3.2.1 Funktsionaalsed nõuded

- **FN1**: Kasutaja saab end autentida mobiilirakendusse, skaneerides genereeritud QR-koodi, mis loob turvalise ühenduse Amphora keskkonnaga.
- **FN2**: Kasutaja saab kasutada automatiseeritud sisselogimist, kui tema seade on juba eelnevalt QR-koodiga seotud ega ole aegunud.
- **FN3**: Amphora veebirakendus kinnitab kasutaja seadme staatust serveris ning hoiab verifitseerimise seisundit 30 minutit või kuni seadme vastuseni.

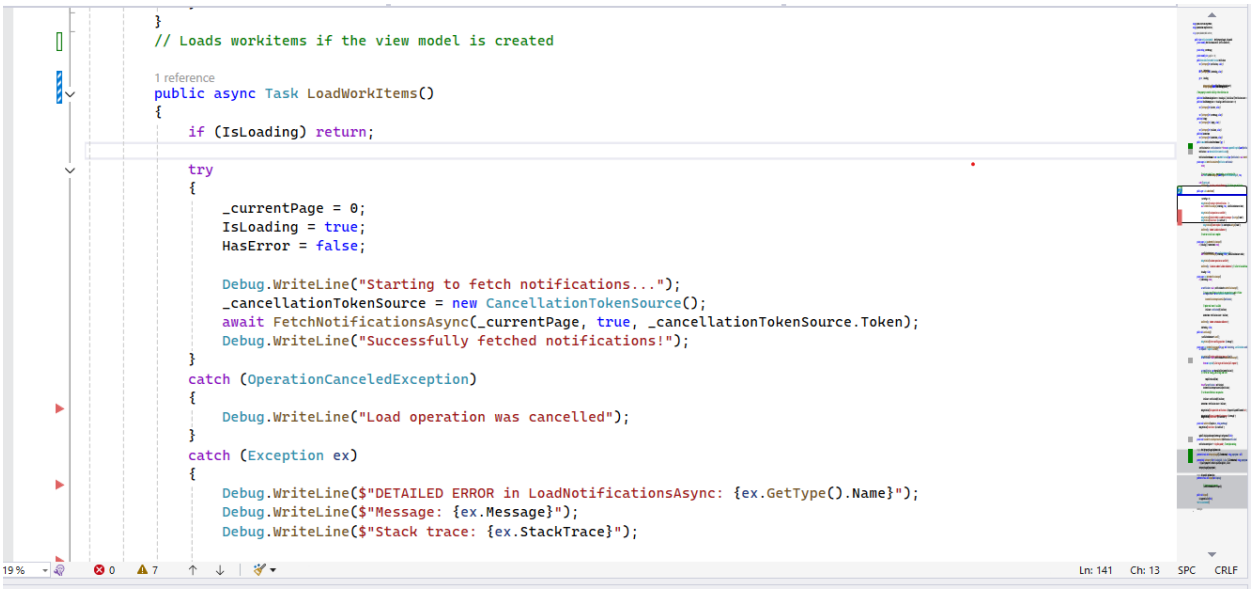
- **FN4**: Kasutaja saab skaneerida QR-koodi, mida Amphora veebikeskkond genereerib ainult eelnevalt veebirakendusse autenditud kasutajale.
- **FN5**: Kasutaja saab iga autentimise jaoks genereerida unikaalse QR-koodi.
- **FN6**: Süsteem säilitab mobiiliseadmes autentimisseisundi ka pärast rakenduse sulgemist.
- **FN7**: Süsteem loob seoseid autenditud seadmete ja Amphora kasutajakontode vahel andmebaasi kaudu.
- **FN8**: Administraator saab andmebaasist vaadata lisatud seadmeid ja nendega seotud infot.
- **FN9**: Süsteem kogub statistikat autentimiskatsete kohta diagnostika ja turvariskide tuvastamise eesmärgil.
- **FN10**: QR-koodi genereerimine veebirakenduses peab toimuma ainult siis, kui kasutaja on autentitud ID-kaardi, Mobiil-ID või Smart-ID abil.
- **FN11**: Tokeni aegumise korral peab kasutaja saama vastava veateate.
- **FN12**: Kasutaja näeb enda Ampora kontol registreeritud mobiilseid seadmeid, mis on lingitud tema Amphora kontoga ning saab eemaldada varasemalt lingitud seadmeid.

3.2.2 Mittefunktsionaalsed nõuded:

- **MFN1**: Rakendus peab toetama nii Android kui ka iOS platvorme.
- **MFN2**: Rakendus peab kohanema erinevate mobiiliseadmete ekraanisuurustega.
- **MFN3**: Rakenduse autentimise funktsioonid peavad toimima koos Amphora veebikeskkonna REST API-ga.
- **MFN4**: Rakendus ja serveripoolsed teenused peavad olema ühilduvad Microsoft SQL Server andmebaasiga.
- **MFN5**: Genereeritud token peab olema võimalikult turvaline.
- **MFN6**: Peale QR-koodi skaneerimist peab 2 sekundi jooksul selguma koodi õigsus.
- **MFN7**: Rakendus ei tohi koormata mobiilseadme protsessorit rohkem kui 0-15%.
- **MFN8**: Süsteem peab logima kõik kriitilised sammud veatuvastuse eesmärgil.
- **MFN9**: Autentimiseks vajalikud andmed peavad säilima seadmes ka rakenduse taaskäivitamisel.
- **MFN10**: Kasutaja saab rakenduse alla laadida Google Play poest või Apple Store’ist.

3.3 Töövahendid

Rakenduse arendamiseks vajalikud ressursid ja vahendid, sealhulgas arendusarvuti, tarkvaralitsentsid ning juurdepääs Amphora veebirakenduse lähtekoodile, tagab Amphora Infohaldus OÜ. Arenduseks kasutatakse Microsoft Visual Studio Community 2022 keskkonda, mis on kasutusel ka firmasiseselt ning sobib hästi .NET MAUI raamistiku ja REST API arenduseks. Joonisel 4 on nähtav arenduskeskkond koos kirjutatud meetodiga tööülesannete jaoks. Projekti koodi haldamiseks kasutatakse Git-i versioonihaldussüsteemi koos GitHubi pilveplatvormiga, kus on võimalik lihtsasti muudatusi jälgida ning vajadusel ka taastada eelnevat versioon. Kood on laetud üles privaatselt vaid valitud kasutajatele ning ei ole avalikult kättesaadav.



Joonis 4. Koodi näide meetodist LoadWorkItems() C# keelest Microsoft Visual Studio keskkonnas.

3.4 Arendus protsess

Rakenduse arendamiseks kasutatakse C# programmeerimiskeelt koos Microsofti .NET MAUI raamistikuga, mis võimaldab luua ühe projekti alusel mobiilirakenduse nii iOS-i kui ka Androidi platvormile. Olulise osa arendusprotsessist moodustab puhta koodi (ingl clean code) põhimõtete järgimine, mille kohaselt peab kood olema lihtsasti loetav, arusaadav ja järjepidev. See saavutatakse näiteks arusaadavate muutujanimede, võimalikult väheste funktsioonide arvuga

ning selge ja struktureeritud koodi kaudu. Samuti välditakse dubleeritud koodi ning eelistatakse koodi, mida on lihtne testida ja hooldada [27]. Lisaks kasutatakse arenduseks Amphora Professional teenusest lokaalselt versiooni, mis võimaldab autoril katsetada süsteemiga tegelemist ilma tagajärgedeta klientidele. Andmekaitse poolest on see samuti vajalik, et autor ei pääseks ligi päris kogu klientide informatsioonile.

Rakendus on üles ehitatud modulaarse arhitektuuri põhimõttel (ingl *MVVM - Model-View-ViewModel*), kus äriloogika ja kasutajaliidese kihid on üksteisest eraldatud. Selline lähenemine aitab parandada koodi loetavust ning lihtsustab tulevikus selle hooldust ja edasi arenduse võimalust. Arenduse käigus pööratakse tähelepanu ka andmekaitsele – mobiilirakendus ei salvesta tundlikku kasutajainfot, järgides seega minimaalse andmesalvestuse nõuet. Lisaks dokumenteeritakse arendusprotsess süsteemselt: iga olulisema klassi ja meetodi juurde lisatakse kommentaarid, mida meetod teeb.

Arendusprotsess toimub iseseisvalt autori poolt määratud tempos ilma kindlate tähtaegadeta. Amphora meeskond on vajadusel valmis aitama tekkinud probleemide ja küsimustega.

3.5 Testimine

Valminud rakenduse testimiseks kasutatakse manuaalset testimist Androidi ja iOS-i seadmetel. Eesmärk on testida kas olemasoleva kasutajakonto sidumine mobiilirakendusega toimib ootuspäraselt ning selle käigus veenduda, et QR-koodi genereerimine ja lugemine, autentimisprotsess ning tööülesannete pärimine töötaks erinevates kasutus olukordades. Peale rakenduse valmimist antakse arendatud rakenduse kood Amhpora arendajatele tagasisidestamiseks ja testimiseks. Testimisprotsessi käigus katsetatakse erinevaid kasutusstsenaariume, alustades rakenduse installeerimisest ja käivitamisest ning seejärel veebirakenduses QR-koodi genereerides. Rakenduse installeerimine toimub läbi arenduskeskkonna, sest avalikult rakendus saadaval töö avaldamise hetkel ei ole. Kontrollitakse kogu töövoogu alates rakenduse avamisest, QR-koodi skaneerimise valimisest kuni tööülesannete lugemiseni. Erilist tähelepanu pööratakse QR-koodiga sidumise protsessi läbimisele erinevates tingimustes, sealhulgas proovimist ka QR-koodiga, mis ei ole Amphora veebirakenduse poolt genereeritud. Tööülesannete loendi laadimisel ja värskendamisel hinnatakse nende vastavust veebirakenduses olevatele andmetele. Samuti testitakse tööülesannete detailvaate avamist ning veendutakse, et kõik andmed kuvatakse

korrektselt ja täielikult. Testimise käigus simuleeritakse erinevaid veaolukordi, nagu aegunud QR-koodi kasutamine või serveri vastuse viivitused, et hinnata rakenduse veakäsitlust probleemsituatsioonides. Kasutajaliidese kohanemisvõimet kontrollitakse erinevate ekraanisuuruste, et tagada kõigile kasutajatele mugav kogemus. Äärmuslike olukordade testimise käigus hinnatakse rakenduse käitumist väga suure tööülesannete arvu korral näiteks 500 ning analüüsitakse pikaajalise kasutuse mõju seadme ressurssidele, et tuvastada võimalikud jõudlusprobleemid.

4. Tulemused

Peatükis annab autor ülevaate teostatud analüüsist ja Amphora infohalduse valminud rakenduse kohta.

4.1 Rakenduste analüüs

Testimisel kasutatakse ainult rakendusi, mis on eestlaste seas kasutusel ning mille kasutajaliides on kas eesti- või ingliskeelne. Testimiseks on välja valitud LHV panga mobiilirakendus, WhatsApp suhtlusrakendus ning Eesti terviseportaal.

4.1.1 LHV rakendus

Apple rakenduste poest ehk *App Store*'ist alla laetud LHV panga mobiilirakendus palub koheselt peale rakenduse avamist siseneda konto kaudu. Rakendus on testimiseks hea, sest kasutab mitmeid erinevaid autentimise lahendusi ning turvalisus on panga jaoks samuti olulisel kohal. Autentimiseks on neli võimalust, kas kasutades Mobiil-ID, Smart-ID, PIN-koodi või varasemalt määratud kasutajanime ja parooli [28]. Sisenedes Mobiil-ID abil on nõutud kaks välja (kasutajanimi ning telefoninumber), mis võimaldab rakendusel suhelda mobiilinumbriga. Peale ekraanil oleva nupu vajutamist, kuvab rakendus kontrollkoodi, seejärel järgneb teavitus, et SIM-kaardile on tulnud teade. Peale teate avamist kuvatakse ekraanile kontrollkood, teenus ja tegevus, mis tuleb samuti kinnitada. Viimane samm on Mobiil-ID koodide sisestamine, keskkonda sisenemiseks on tavaks küsida kasutajalt esimest koodi. Peale koodi sisestamist lubab rakendus kasutajal panka siseneda. Teine viis on kasutada Smart-ID sisenemist, mis ei sisalda kuumakset võrreldes Mobiil-ID-ga. Testiks on kasutusel uuesti LHV mobiilirakendus, esialgne sisenemise vaade erinevalt eelnevalt mainitud vaatest, on telefoninumbri asemel sisenemiseks vajalik isikukood, mida kasutab Smart-ID. Peale väljade täitmist kuvab rakendus uuesti kontrollkoodi ekraanile, seejärel saadab Smart-ID rakendus teate, et kinnitamist ootav protsess on saabunud. Seejärel kuvab Smart-ID rakendus kolm koodi, millest tuleb valida LHV poolt antud kontrollkood ning viimaseks sisestada Smart-ID seadistamisel valitud parool ja sisenemiseks on tavaks kasutada PIN 1. Sisenemise tühistamisel on lihtne ja kasutajasõbralik lähenemine, katkestamiseks loodud punast nuppu vajutades viib rakendus tagasi esimesse vaatesse, kus on saadaval erinevad sisenemise võimalused. Ilma lisasammudeta parooli ja kasutajanime sisestades tuleb kuuendal

katsel teade, et sisenemiskatsete tõttu on ajutiselt salasõna blokeeritud ehk kaitstud jõurünnaku eest. Erinevate lahenduste eeliseks on näiteks mõne autentimise viisi puudumisel võimalik kasutada alternatiive, kuid ka tehniliste probleemide tõttu on vajadusel olemas mitu viisi kasutajal siseneda. Teine koht, kus kasutatakse kaheastmelist autentimist, on veebikeskkonnad, mis ei vaja rakenduse allalaadimist.

4.1.2 Eesti terviseportaal

Eesti terviseportaal on koht, kus on kodaniku isiklikud meditsiinilised andmed ning kõik isikuga seonduvad andmed tervishoiusüsteemis [29]. Tegu on väga delikaatsete andmetega ja turvarikke puhul oleks tagajärjed tõsised, selle tõttu polegi terviseportaalil võimalust keskkonda siseneda kasutajanime ja parooli või mõne PIN-kalkulaatori abil. Probleem kasutajanime ja parooliga on, et vabadus ise parool valida tekitab kasutajale võimaluse valida lihtne parool, mida on lihtsam pahatahtlikul eesmärgil ära arvata. Hea tava oleks parooli hoida vaid peas või paroolihalduris, sest laual olevat paberit võivad näha kõrvalised isikud ja saada selle abil ligipääsu kasutajale [30]. Seetõttu ongi terviseportaali võimalik siseneda vaid ID-kaardi, Mobiil-ID ja Smart-ID abil, see vähendab inimlikku eksimise võimalust ning jõurünnaku tõhusust või üldse võimalust. Erinevus mobiilirakendusega on ka ID-kaardiga autentimine, sest enamik arvutitest toetab kaardilugeja kasutust ning kaasaskantavatel seadmetel, välja arvatud sülearvutil, see võimalus antud hetkel veel puudub. ID-kaardiga autentimiseks on vajalik kasutaja isikukood, DigiDoc ID-tarkvara ning kaardiga tulnud PIN-koodid, mida on kaks. Esimene tavaliselt sisenemiseks ja teine allkirjastamiseks, näiteks tehingud, lepingud jne. ID-kaardiga autentimise protsess on eelnevatele üsnagi sarnane, peale isikukoodi sisestamist kuvab veebilehitseja, mis on rakendus arvutis või nutiseadmes, mille eesmärgiks on veebilehtede vaatamise võimaldamine. Veebilehitseja tekitab akna, kus on valik kaardilugejas olevatest kaartidest ning seejärel kuvab aken lahtri, kus on väli PIN-koodi sisestamiseks. See on turvaline viis autentimiseks, sest kasutamiseks on vajalik rohkem kui vaid salasõna ja kasutajanimi.

4.1.3 WhatsApp rakendus

WhatsApp on suhtlus rakendus, mis kasutab QR-koodi põhist autentimist WhatsApp Webi ehk veebiseanssi aktiveerimiseks. Rakenduse kasutaja, kes soovib arvutist oma vestlustele ligi pääseda, avab veebilehe, kus kuvatakse ajutine ja unikaalne QR-kood. Selle koodi skaneerimiseks

avab kasutaja oma nutitelefonis WhatsApp mobiilirakenduse ning kasutab sisseehitatud QR-koodi skannerit. Skaneerimise tulemusena luuakse krüpteeritud ühendus mobiilirakenduse ja arvuti vahel, võimaldades kasutajal kiiresti ja turvaliselt seanssi aktiveerida ilma täiendava paroolisisestuseta. Turvalisuse tagamiseks toimib autentimine juba eelnevalt kinnitatud mobiilirakenduse kaudu. Kuvatavat QR-koodi genereeritakse teatud aja tagant uuesti, see selle raskesti ära kasutatavaks pahalaste jaoks. Samuti toetub kogu WhatsApp sõnumivahetus end-to-end krüpteerimisele, mis hoiab kasutajate isiklikud ja konfidentsiaalsed andmed turvalisena [22].

4.2 Rakenduse implementeerimine

Amphora mobiilirakenduse arenduse käigus on loodud mobiilirakendus, mis võimaldab kasutajatel mugavalt ligi pääseda infohaldussüsteemi tööülesannetele. Järgnevates alapeatükkides kirjeldatakse detailselt rakenduse struktuuri, kasutajaliidest, autentimisprotseduure ja andmevahetuse mehhanisme.

4.2.1 Arenduse käigus täidetud nõuded

Enne töö algusest sai määratud funktsionaalsed ja mittefunktsionaalsed nõuded. Tabelites 1 ja 2 on näha, et enamik nõudeid õnnestus arenduse käigus edukalt realiseerida. Saavutatud tulemused loovad hea aluse rakenduse edasiseks arendamiseks, tagades juba praegu põhifunktsionaalsuse toimimise.

Tabel 1. Funktsionaalsete nõuete täitmine

Funktsionaalne nõue	**Kirjeldus**	**Täidetud**	**Täpsustused**
FN1	QR-koodi põhine autentimine	Jah	Rakendusse on võimalik autentida QR-koodi abil
FN2	Automaatne sisselogimine	Jah	Toimib, kui token pole aegunud
FN3	Verifitseerimise hoidmine	Jah	30-minutiline kehtivusaeg implementeeritud
FN4	QR-koodi genereerimine veebirakendusse autenditud kasutajale	Jah	Implementeeritud veebirakenduse poolel
FN5	Unikaalse QR-koodi genereerimine	Jah	Iga QR-kood on unikaalne
FN6	Autentimisseisundi säilitamine	Jah	Säilib seadme mälus
FN7	Seadmete ja kontode sidumine	Jah	Implementeeritud andmebaasis
FN8	Seadmete haldamine administraatori poolt	Jah	Nähtav andmebaasi vaates
FN9	Statistika kogumine	Jah	Logitakse autentimiskatsed
FN10	QR-koodi turvaline genereerimine	Jah	Eeldab turvalist eelautentimist
FN11	Aegunud tokeni veateade	Jah	Kasutajale kuvatakse selge veateade
FN12	Registreeritud seadmete nägemine ja haldamine	Ei	Ei mahtunud skoopi

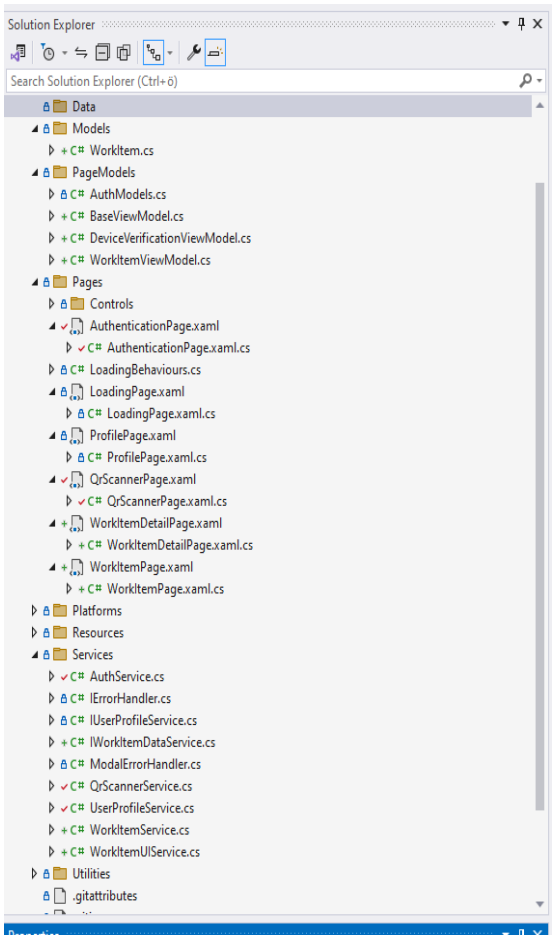
Tabel 2. Mittefunktsionaalsete nõuete täitmine

Mittefunktsionaalsed nõuded	**Kirjeldus**	**Täidetud**	**Täpsustused**
MFN1	Android ja iOS tugi	Jah	Mõlemad platvormid toetatud
MFN2	Ekraanisuurustega kohanemine	Jah	Kohaneb erinevate seadmetega
MFN3	REST API integratsioon	Jah	Rakendus ja Server kasutavad päringute jaoks REST API
MFN4	MS SQL Server ühilduvus	Jah	Andmebaasiühendus toimib
MFN5	Genereeritud token turvaline	Jah	Genereeritakse piisavalt pikk tagasüsteemis
MFN6	QR-koodi verifitseerimise kiirus	Jah	QR-koodi õiguses veendutakse 2 sekundi jooksul peale QR-koodi tuvastamist
MFN7	Ressursitõhusus	Jah	Rakendus ei kasuta liigselt resurssi
MFN8	Logimine	Jah	Vajadusel logib rakendus kogu nõutud info
MFN9	Ligipääsu säilimine peale rakenduse sulgemist ja uuesti avamisel	Jah	Rakendus võimaldab ilma uuesti autentimata siseneda, kuni token kehtib
MFN10	Rakenduse kättesaadavus	Ei	Rakendus on testimisjärgus

4.2.2 Amphora mobiilirakenduse struktuur

Antud arhitektuurimustri (ingl Model) ehk andmemudeli komponendis paikneb rakenduse äriloogika ja andmestruktuurid, mis funktsioneerivad sõltumatult kasutajaliidesest ja selle operatsioonidest. (ingl *View*) ehk kasutajaliidese komponent vastutab andmete visuaalse esitamise eest kasutajale ning kasutaja tegevuste edastamise eest (ingl *ViewModel*). Antud vahekiht toimib integreeritud komponendina mudeli ja kasutajaliidese vahel, pakkudes andmesidumise võimalusi, mille kaudu kasutajaliidese komponent pääseb ligi vajalikele andmetele ja funktsioonidele. See

arhitektuuriline disain võimaldab vahekihil efektiivselt kommunikeerida andmete muutustest kasutajaliidesele ning hallata kasutaja interaktsioone süsteemiga [25]. Rakenduse ja Amphora tagasüsteemi (ingl *back-end*) failide struktuur on nähtav Joonisel 5. Projektis on selgelt eristatavad kihid:



Joonis 5. Käesoleva rakenduse failide struktuur.

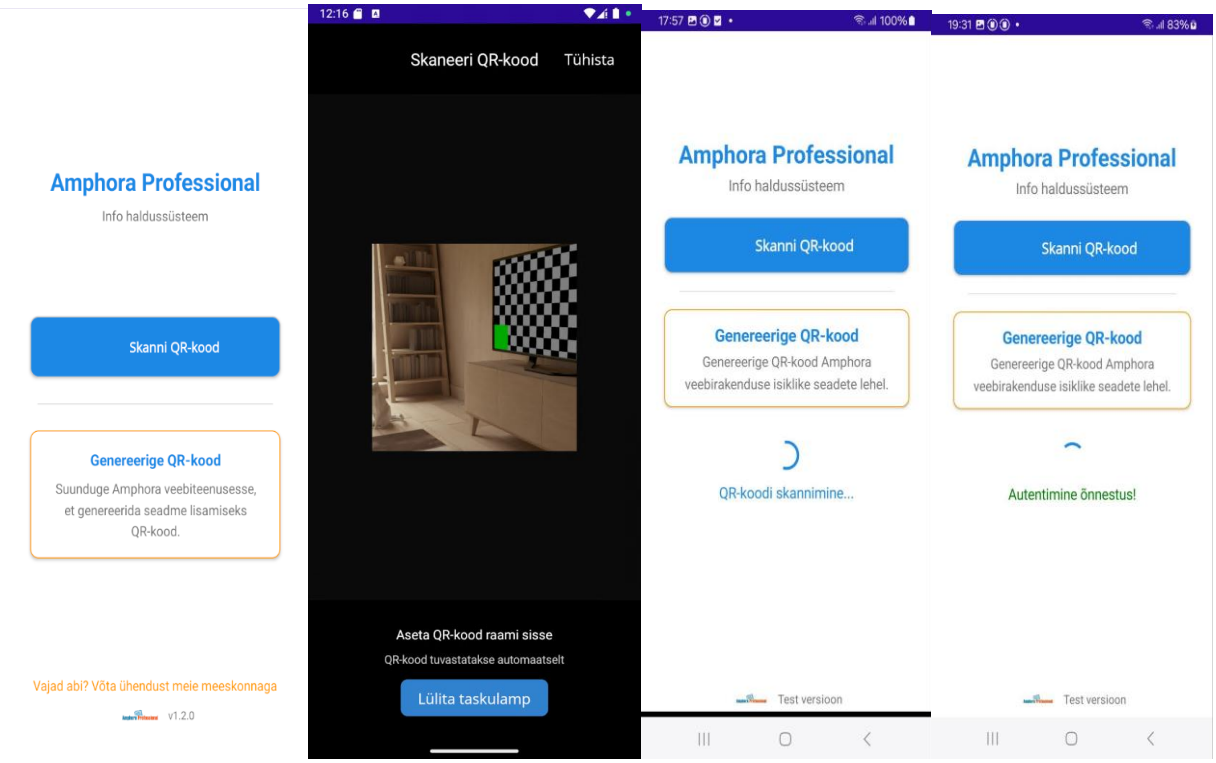
1. Rakenduse poolel Models kaust sisaldab rakenduse andmemudelit WorkItem, mis määratleb tööülesannete andmestruktuuri ja peegeldab Amphora andmebaasi disaini.
2. PageModels kaust hõlmab vaadete mudeleid (ViewModels), mis toimivad vahekihina kasutajaliidese ja andmemudelite vahel, koordineerides andmevoo ja kasutajaliidese loogikat.

3. Pages kaust koosneb XAML-märgenduskeeles loodud kasutajaliidese vaadetest, mis visualiseerivad rakenduse erinevaid osasid. Samuti ka xaml.cs, kus on võimalik muutujaid defineerida.
4. Services kaust sisaldab klasse, mis vastutavad andmevahetuse eest taustasüsteemidega, sealhulgas Amphora API-ga.

4.2.3 Kasutajaliides

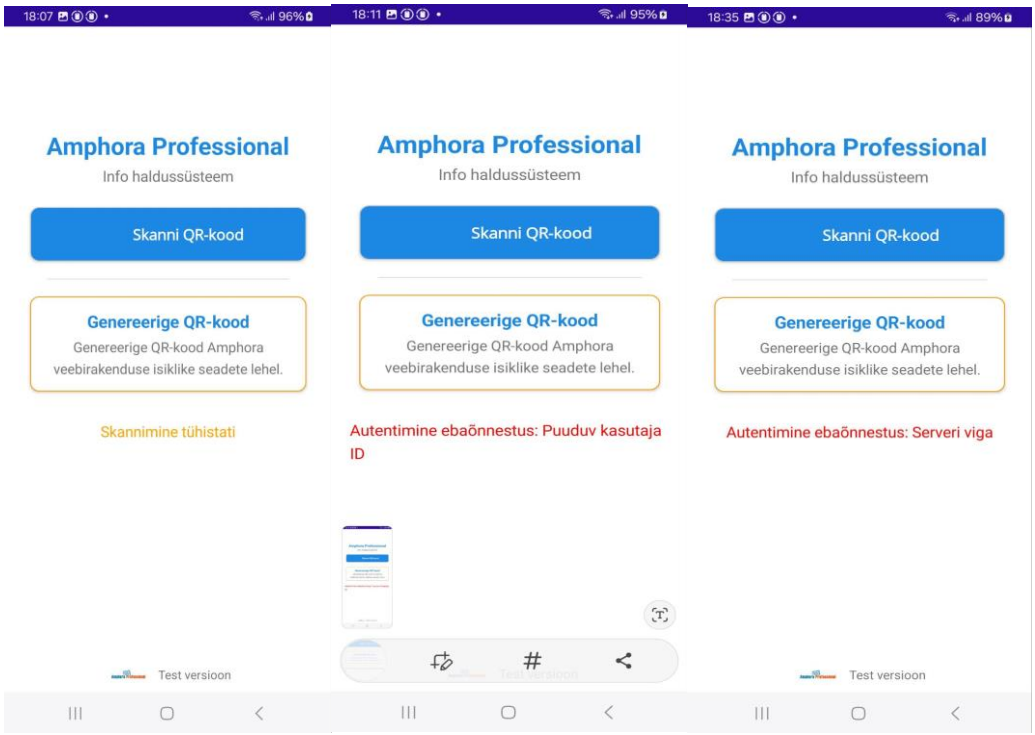
Rakenduse kasutajaliides on loodud võimalikult kasutajasõbralikuks ja lihtsaks. Kasutajaliidese loomisel oli abiks analüüsis testitud rakendused, kust oli näha sagedasi jooni, kuhu paigutatakse ekraanil nupud. Loodud on järgmised põhivaated:

Autentimisvaade ja **QR-Koodi** skaneerimine ning edukas autentimine, mis on nähtavad jooniselt 6. Joonisel 6a on kasutajal võimalik valida QR-koodi skaneerimine koos juhendiga, kuidas vastav QR-kood leida. Vaade on disainitud minimalistlikult, keskendudes QR-koodi skaneerimise funktsionaalsuse esile toomisele ning juhendama kasutajat, kust QR-kood genereerida. Joonisel 6b on nähtav QR-koodi skaneerimise vaade, mis enne küsib seadmelt luba kasutada kaamerat ning seejärel palub kaamera suunata QR-koodile, mille rakendus tuvastab automaatselt. Juhuks kui tegemist on pimeda valgustusega pakub rakendus võimalust kasutada seadme välku, mis võib parandada pildikvaliteeti. Kui paremalt ülevalt nurgast valida tühista valik siis väljastab rakendus teate, mis on nähtav jooniselt 6c. Eduka autentimiskatse järel kuvab hetkeks rakendus teate “Autentimine õnnestus!”, nähtav jooniselt 6d.



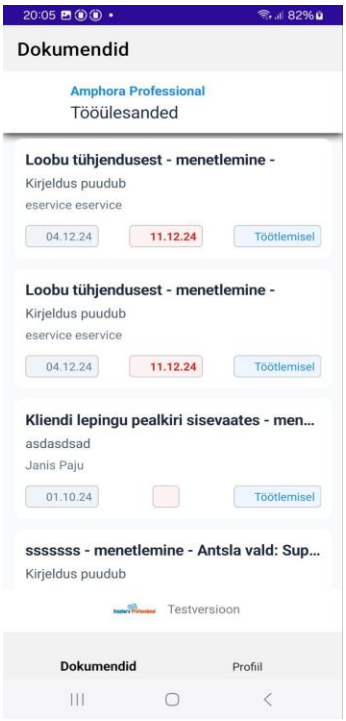
Joonis 6. Kasutajaliidese vaated: (a) rakenduse avavaade, (b) QR-koodi skaneerimine, (c) QR-koodi skaneerimine ja (d) edukas autentimine.

Rakendus pakub kasutajale selgeid ja informatiivseid veateadete kuvamisi autentimise protsessi erinevates etappides, mis on nähtavad jooniselt 7. Veateated on disainitud nii, et need annaksid kasutajale selget tagasisidet esinenud vea kohta. Joonisel 7a on näha olukord, kus QR-koodi skaneerimine on katkestatud, mille korral rakendus kuvab teate "Skaneerimine tühistati". Joonisel 7b on kujutatud autentimise ebaõnnestumist puuduva kasutaja ID tõttu. Veateade "Autentimine ebaõnnestus: puuduv kasutaja ID" annab selge indikatsiooni, et skaneeritud QR-kood ei sisaldanud vajalikku teavet või oli see vigane. See aitab kasutajal mõista, et probleem võib olla QR-koodi genereerimises. Joonisel 7c on näha serveripoolse vea kuvamine kasutajale. Veateade "Autentimine ebaõnnestus: serveri viga" teavitab kasutajat, et probleem ei ole tema seadmes ega tegevustes, vaid serveripoolses komponendis. Kõikide veateadete puhul on kasutajal võimalus naasta rakenduse avalehele, et autentimist uuesti proovida, või vajadusel võtta ühendust süsteemi administraatoriga täiendava abi saamiseks.



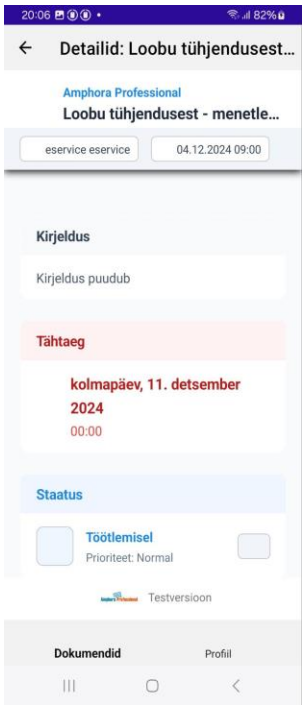
Joonis 7**.** Veateated autentimisel: (a) skaneerimise tühistamine, (b) puuduv kasutaja ID, (c) serveripoolne viga.

Peale edukat keskkonda sisenemist avaneb tööülesannete loendvaade, mis on nähtaval joonisel 8. Vaade kuvab ekraanil kasutaja tööülesanded kompaktses ja loomise järjekorras, eespool on uuemad ülesanded. Lohistades ülesandeid ekraanil alla aktiveerub värskendusfunktsioon, mis kuvab ekraanile laadimise animatsiooni ning teeb taustal uue päringu teadete uuendamiseks.



Joonis 8. Tööülesannete loendvaade rakenduses, kuvamas kasutaja tööülesandeid.

Joonisel 9 on nähtav ülesannete detailvaade, mis kuvab loendist valitud tööülesanded täpsemat informatsiooni kogu ekraani ulatuses, võimaldades kasutajal tutvuda ülesande detailidega.



Joonis 9. Tööülesande detailvaade, näitamas põhjalikku informatsiooni.

4.2.4 Andmevahetus ja andmehaldus

Rakenduse andmevahetus Amphora serveriga toimub kasutades mobiilirakenduse klasse:

- WorkItemService - kasutab REST API teenust WorkItemProvider
- AuthService - kasutab REST API teenust Verify
- QrScannerService

Amphora veebirakenduse poolel on REST API teenused:

- Verify (Verify.cs)
- WorkItems (WorkItemsController.cs)

WorkItemService tegeleb tööülesannete vahetuse ja haldamisega. Klass võimaldab kasutajal näha veebirakenduse kaudu suunatud tööülesandeid. Ühendus serveriga toimub REST API kaudu, kus teenus teeb päringuid vastavalt käskudele, näiteks värskendamine.

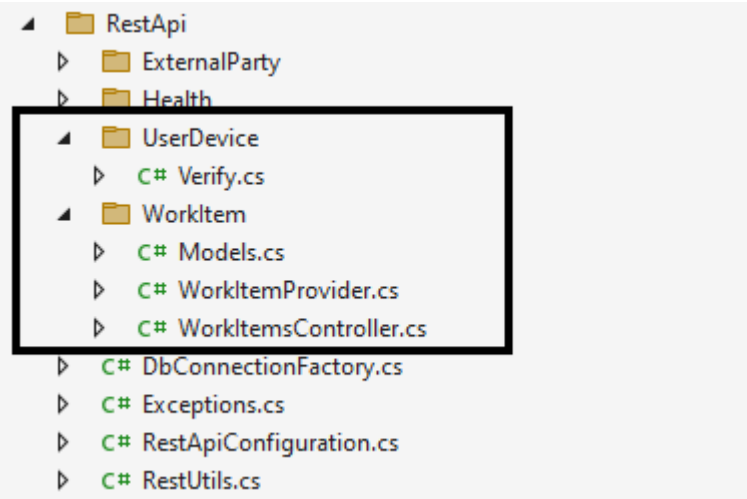
AuthService klassis on antud mobiilirakenduse autentimise komponent, mis vastutab kasutaja tuvastamise, turvalisuse ja seadme autentimise eest. See klass toimib vahendajana mobiilirakenduse ja Amphora serveri vahel, tagades, et ainult volitatud kasutajad pääsevad ligi rakenduse funktsionaalsusele. Autentimine toimub QR-koodi skaneerimise kaudu, kus kasutaja skanneerib Amphora veebiliidesest genereeritud QR-koodi, mis sisaldab autentimiseks vajalikke andmeid nagu token, konto ID ja API lõpp-punkti (ingl *end-point*) URL. Andmed, mida saadab veebirakendus on *JSON* kujul ning nähtavad jooniselt 10. Lisaks analüüsib klass skannitud QR-koodi, eraldab ja võtab *JSON* kujust vajalikud andmed ja saadab need tagasi serverisse verifitseerimiseks.

Joonis 10. *JSON* test päring Amphora veebirakenduselt.

QrScannerService on rakenduse komponent, mis vastutab QR-koodide skaneerimise funktsionaalsuse eest. See teenus moodustab olulise ühenduslüli rakenduse autentimisprotsessi ja kasutajaliidese vahel. Teenuse põhiline ülesanne on hallata kaamerapõhist QR-koodide skanneerimist, alustades kaamerapääsu lubade kontrollimisest ja taotlemisest. Teenus tagab, et kasutajalt küsitakse asjakohaseid õigusi, ilma milleta skaneerimise funktsionaalsus ei ole võimalik. Kaamera lubade haldus järgib MAUI platvormide õiguste turvamudelit, mis tagab kasutajatele läbipaistva kontrolli rakenduse ressursside üle. QR-koodi skaneerimise protsessi ise on *ScanQrCode* meetodis, mis käivitab *QrScannerPage* kasutajaliidese. Kui QR-kood on edukalt skaneeritud, jõuab tulemus läbi sündmuse *QrCodeScanned* meetodini, mis sulgeb skanneri vaate ja edastab tulemuse protsessile *TaskCompletionSource*. See võimaldab muuta keeruka UI-põhise interaktsiooni lihtsaks protsessiks, mida rakendus saab kasutada.

Verify on Amphora serveripoolne teenus, mis moodustab keskseks ühenduspunktiks rakenduse autentimisprotsessis. ASP.NET Web API kontroller vastutab kasutajate seadmete autentimise, valideerimise ja registreerimise eest serveripoolel ehk see meetod algatab andmebaasi protseduuri.

Põhifunktsionaalsus seisneb Post meetodis, mis toetab kaht erinevat toimingut: uue seadme registreerimist QR-koodi skaneerimise kaudu ja olemasoleva seadme tokeni valideerimist. Vastavalt parameetritele (*token*, *userId* ja *isQrScan*) suunatakse päring kas *CreateNewUserDevice* või *VerifyExistingDevice* meetodisse. Joonisel 11 on nähtav serveripoolne failide struktuur.



Joonis 11. Amphora tagasüsteemis olevad failid, mis on kasutusel rakendusega suhtlemiseks on tähistatud musta kastiga.

Uue seadme registreerimise protsessis (*CreateNewUserDevice*) genereeritakse unikaalne token, seatakse kehtivusaeg esialgu 30 minutit, mille sees oodatakse vastust seadmelt ning salvestatakse informatsioon andmebaasi *Add_UserDevice* protseduuri kaudu. Eduka registreerimise korral tagastatakse seadmele kasutaja ID, token ja selle aegumiskuupäev, mida mobiilirakendus kasutab edaspidistes päringutes. Olemasoleva seadme valideerimise protsessis (*VerifyExistingDevice* meetod) kontrollitakse tokeni kehtivust ning uuendatakse seadme staatust *Update_UserDevice* protseduuri kaudu. Lisaks kontrollitakse tokeni aegumiskuupäeva, tagastades vastava veateate, kui token on aegunud või seda ei leitud. Turvalisuse tagamiseks rakendatakse põhjalikku veakäsitlust *try-catch* plokkide abil, mis tagab, et potentsiaalsed vead on korrektselt käsitletud ja kliendile on esitatud asjakohane tagasiside.

WorkItemController on Amphora serveripoolne klass, mis koos *DatabaseWorkItemProvider* implementatsiooniga moodustab tööülesannete halduse tagasüsteemis. See komponent vastutab tööülesannete pärimise, töötlemise ja staatuse uuendamise eest, toimides sillana andmebaasi ja REST API vahel. Lisaks on kontrollid (*MinCheckInterval*) ja päringu ajalimiidi

kontroll(*CheckTimeout*), mis aitab vältida liigset koormust andmebaasile. *FetchWorkItems* meetod koordineerib kogu tööülesannete pärimise protsessi. *DatabaseWorkItemProvider* implementeerib spetsiifilise andmebaasipõhise tööülesannete allika, mis suhtleb SQL Serveri andmebaasiga.*RunWorkItemQuery* meetod täidab salvestatud protseduuri "*Get_ProcessTasksForDelegationPage*", mis võtab tööülesanded andmebaasist vastavalt kasutaja ID-le. Kasutaja ID tuvastatakse tokeni abil, mis tuleb iga päringuga kaasa. Iga päringu raames kontrollitakse tokeni vastavust ja kehtivust. Lisaks iga andmebaasist päritav tööülesane teisendatakse *WorkItem* ehk tööülesande mudeliks, mis sisaldab olulist infot nagu pealkiri, kirjeldus, prioriteet, tähtaeg ja looja, *WorkItem* mudel on nähtav jooniselt 12.

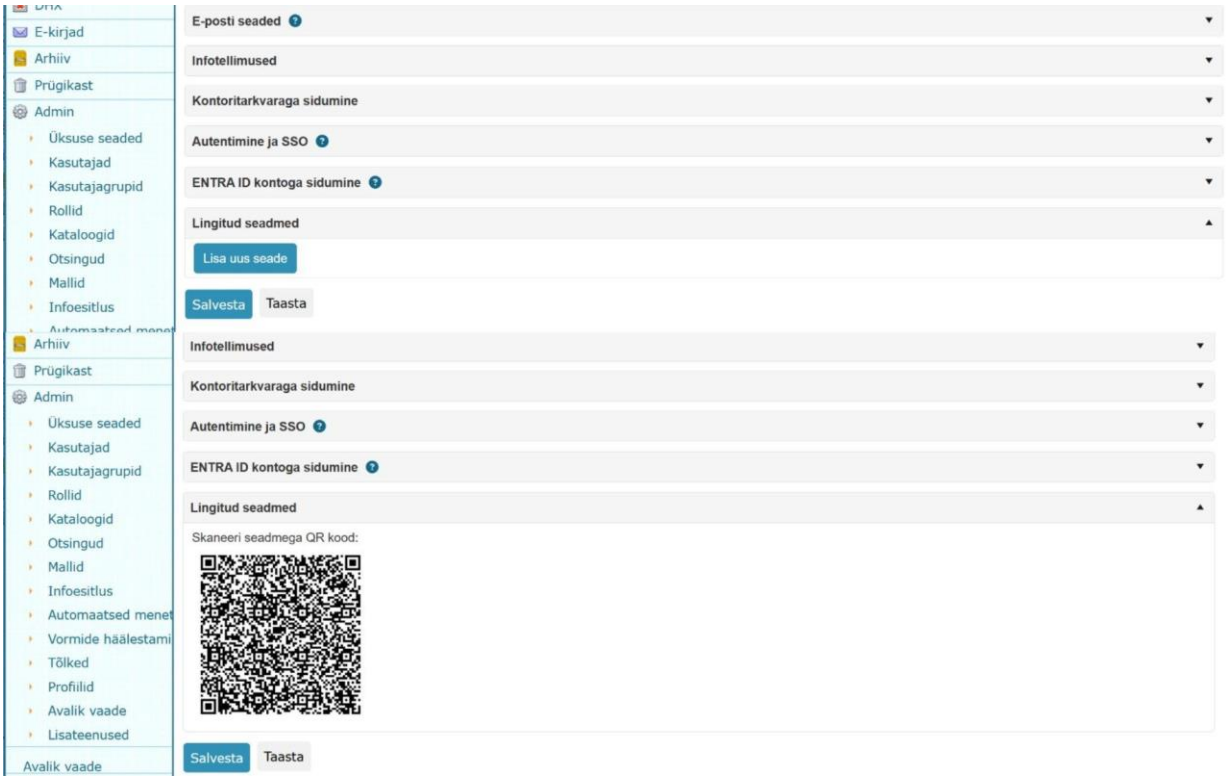
```
11 references
public class WorkItem
{
    2 references
    public long WorkItemId { get; set; }
    1 reference
    public string Title { get; set; }
    1 reference
    public string Priority { get; set; }
    1 reference
    public DateTime? DueDate { get; set; }
    1 reference
    public bool Seen { get; set; }
    1 reference
    public DateTime CreatedDate { get; set; }
    1 reference
    public int StatusId { get; set; }
    1 reference
    public string Description { get; set; }
    1 reference
    public string Creator { get; set; }
    1 reference
    public string AssignedPerson { get; set; }
    2 references
    public int TotalCount { get; set; }
}
```

Joonis 12. Tööülesannete mudel.

4.2.5 Autentimismeetod

Valminud autentimissüsteemi üks olulisemaid turvaaspekte on tokeni käsitlemine. Iga genereeritud token läbib mitu olulist etappi. Esiteks, pärast QR-koodi loomist ja tokeni genereerimist, salvestatakse token algselt andmebaasi mitteaktiveeritud staatuses. QR-koodi loomine Amphora veebirakenduses on nähtav jooniselt 13. QR-koodi genereerimine on esialgselt

lisatud Amphora lokaalsesse kasutajaliidesesse personaalsete seadete alla. QR-koodi genereerimise võimalus veebirakendusse loodi Amphora arendajate poolt, et veenduda paigutuses, et genereerimise asukoht oleks loogiline ka ülejäänud süsteemiga. QR-koodi koostamise käigus algatatakse baasis uue seadme lisamise protsess käivitades protseduuri *Add_UserDevice* (sisaldab ka QR-koodis olevat tokenit), mis lisab *sys_users_device* tabelisse kirje, kus on kasutaja ID, token, loomise aeg ning tokeni aegumise aeg. Seejärel, kui kasutaja skaneerib QR-koodi oma seadmega, tuvastab rakendus QR-koodist tokeni ning käivitab verifitseerimise protsessi. Verifitseerimise käigus kontrollitakse tokeni aegumist ja vastet Amphora andmebaasis. Kui token on kehtiv, märgitakse see aktiveerituks läbi *Update_UserDevice* andmebaasi protseduuri, mis muudab tokeni aktiveerituks ja salvestab aktiveerimise aja ning pikendab tokeni kehtivust kolme kuu võrra.



Joonis 13. Amphora veebirakendus QR-koodi genereerimine ja ilmumine. Ülemine osa pildist on kasutaja vaade kui soovitakse seade lisada. Alumine osa on peale nupu vajutamist genereeritud QR-kood.

4.2.6 Turvalisus

Amphora mobiilirakenduse sidumine toimub järgmiste sammude kaudu:

1. **Eelautentimine:** Amphora veebikeskkond nõuab nutiseadme sidumiseks kasutajalt esmast autentimist veebirakendusse kõrge turvatasemega meetoditel (ID-kaart, Mobiil-ID, Smart-ID või GovSSO), mis võimaldab kasutaja identifitseerida.
2. **Kasutaja valideerimine:** Peale QR-koodi genereerimist kasutab mobiilirakendus serveri poolt saadud unikaalset kasutaja ID, mis võimaldab probleemi või kahtluse korral näha, millise kasutaja QR-koodi on skaneerimisel kasutatud. Kui ühe kasutaja seadmete arv tõuseb turvarikke korral liialt kõrgeks on see samuti lihtsasti jälgitav.
3. **Seadmete sidumine:** Iga seade registreeritakse süsteemis unikaalse identifikaatoriga ning seotakse kindla kasutajaga, mis salvestatakse SQLi andmebaasi tabelisse *sys_users_device*. See võimaldab kasutajal kui ka administraatoril hallata seadmeid ja vajadusel tühistada ligipääs konkreetsetele seadmetele. Näiteks kui tekib turvarike, kus kasutajale lisatakse 10 seadet juurde, siis on tabeli abil võimalik näha, milline oli kasutaja originaalne seade või seadmed.

4.2.7 API turvalisus

REST API lõpp-punktid (ingl *end-point*) on implementeeritud samuti turva kaalutlusi silmas pidades. Tagasüsteemis olev *Verify ja Workitems* teenused kasutavad valideerimisi, hõlmates sisendandmete kontrolli, tokeni olemasolu ja aegumise valideerimist. Samuti väljastatakse veateateid kliendile nii, et vaid vajalik informatsioon kuvatakse klientidele, vältides potentsiaalseid turvaauke, mis võiksid tekkida liiga detailsete veateadete tagastamisel.

4.2.8 Rakenduse jälgimine

Süsteemis on võimalusi rakenduse tegevuste ja sammude jälgimiseks erinevatel tasemetel. Näiteks andmebaasis salvestatakse iga tokeni aktiveerumise aeg, mis võimaldab tagantjärele analüüsida autentimise katseid ja tuvastada võimalikke sisenemise katseid. Mobiilirakendus kuvab arendaja konsoolis olulised sündmused kasutades Debug.Writeline käsku, mis kuvab õnnestunud samme ja võimalikke probleeme. Joonisel 14 on rakenduse koodist näide, kus rakendatakse erinevaid samme, mis võimaldavad arendajal jälgida mida täpselt rakendus teeb. Need teated sisaldavad

ajahetki, veateateid ja operatsioonide tulemusi, võimaldades süsteemi täiendavat lihtsasti edasi arendada. Samuti kantakse tulevikus info Amphora logimissüsteemi.

```
using (var client = new HttpClient(handler))
{
    Debug.WriteLine("Created HTTP client with custom handler");
    client.Timeout = TimeSpan.FromSeconds(30);
    Debug.WriteLine("Sending HTTP request...");
    var response = await client.GetAsync(_apiUrl);
    Debug.WriteLine($"Received response with status code: {response.StatusCode}");
    var content = await response.Content.ReadAsStringAsync();
    Debug.WriteLine($"Response content length: {content.Length} characters");
    Debug.WriteLine($"First 100 chars of response: {content.Substring(0, Math.Min(100, content.Length))}");

    if (response.IsSuccessStatusCode)
    {
        Debug.WriteLine("Attempting to deserialize content...");
        var workItemResponse = System.Text.Json.JsonSerializer.Deserialize<WorkItemListResponse>(content);

        if (workItemResponse != null && workItemResponse.WorkItems != null)
        {
            Debug.WriteLine($"Successfully deserialized {workItemResponse.WorkItems.Count} notifications");
            _cachedNotifications = workItemResponse.WorkItems;
            _lastCacheTime = DateTime.Now;
            return _cachedNotifications;
        }
```

Joonis 14. Rakenduse koodis abiks olevad logimise viisid.

5. Arutelu

Käesolevas peatükis arutletakse lõputöö käigus saavutatud tulemuste, kogetud väljakutsete ning nende lahenduste üle. Samuti esitatakse soovitusi Amphora infohaldussüsteemi mobiilirakenduse edasiseks arendamiseks.

5.1 Autentimise lahendused Eestis

Eesti on digitaalsete lahenduste kasutamises olnud rahvusvaheliselt eeskujuks teistele riikidele. Mitmetasemelise autentimise valdkonnas on Eestis kasutusel mitmeid lahendusi, mis on märkimisväärselt edukamad kui paljudes teistes riikides. Eesti digitaalse identiteedi lahendused nagu ID-kaart, Mobiil-ID ja Smart-ID on pakkunud kodanikele turvalist ja usaldusväärset ligipääsu e-teenustele juba mitmeid aastaid. Nende lahenduste edukuse võtmeteguriteks on olnud riiklikult tagatud infrastruktuur, e-teenuste laiapõhjaline kasutatavus nii avalikus kui ka erasektoris ning pidev tehnoloogiline innovatsioon. Uurides Eestis kasutatavaid autentimismeetodeid, on näha, et need põhinevad tugeval kaheastmelisel autentimisel, mis pakub oluliselt paremat turvataset võrreldes tavaliste kasutajanime-parooli lahendustega. Kaheastmeline autentimine on Eestis muutunud standardiks nii riiklikes e-teenustes kui ka erasektoris, näiteks internetipanganduses, millest annab tunnistust ka töös testitud LHV panga rakenduse autentimissüsteem [21]. Eesti terviseportaali analüüs näitab samuti, et delikaatsete andmete puhul pööratakse erilist tähelepanu turvalisusele, võimaldades ligipääsu ainult tugeva autentimisega [22].

Vaatamata edusammudele esineb mitmetasemelise autentimise rakendamisel Eestis siiski mõningaid väljakutseid. Üheks selliseks on kasutajate mugavuse ja turvalisuse tasakaalustamine. Turvalisemate lahenduste, näiteks ID-kaardi ja kaardilugeja kasutamine, võib olla ebamugavam kui lihtsamad lahendused. Teiseks väljakutseks on teenusepakkujate poolne vastuseis – mõned organisatsioonid, sealhulgas Amphora Infohaldus Oü, kaaluvad turvalisemaid autentimismeetodeid (Smart-ID) lisatasustada, sest Smart-ID kasutamine võib osutuda teenusepakkujale kulukaks, seega võib see piirata antud lahenduse laiemat kasutuselevõttu. Kolmandaks võivad tehnilised barjäärid, nagu spetsiifiliste seadmete vajadus või tasulised teenused, saada takistuseks osadele kasutajatele. Võrdluses teiste riikidega on Eesti autentimislahendused siiski märkimisväärselt turvalised.

5.2 Rakenduse arendus

Käesoleva lõputöö raames Amphora mobiilirakenduse arendamine oli protsess, mille käigus tuli varasemalt teostatud analüüsi käigus kogutud teadmised ja lahendused rakendada praktikasse ning viia kooskõlasse firma poolt seatud nõuetega. Näiteks kui oluline on õiges kohas veateateid väljastada, et vältida olukorda, kus rakendus lihtsalt sulgub mitte ühegi teavituseta ning kasutaja ei satuks segadusse.

Analüüsis teostatud WhatsAppi QR-koodiga autentimise testimine osutus eriti kasulikuks eeskujuks, kuna see pakkus lihtsaimat kasutajakogemust ilma täiendavaid andmeid sisestamata. Antud lahenduse puhul on mugav, et on võimalik nutiseade ning arvuti siduda lihtsasti - piisab vaid rakenduse ja õige menüü avamisest ning QR-koodi skaneerimisest. QR-koodi kasutamine on hea valik, eriti arvestades, et Amphora kasutajad on juba eelnevalt veebirakendusse autenditud, mis tagab esmase turvatõkke. Lisaks analüüsides LHV panga mobiilirakendust selgus vajadus täpsete ja selgete veateadete järele. Eriti oluliseks saigi selle tulemusena rakenduses autentimise või tööülesannete pärimise ebaõnnestumise korral tagasiside andmine kasutajale - kas probleem on näiteks aegunud token, vigases QR-koodis, tööülesannete puudumine või serveripoolses veas. Täidetud said enamik ette seatud funktsionaalsed ja mittefunktsionaalsed nõuded, kuid nõuded FN12 ja MF10, jäid kahjuks täitmata. FN12 on võimalik teostada, tulevikus kui rakendus on valmis avalikuks minema ja Amphora reaalajas versioon saab ka selle täienduse. Hetkel on võimalik ka kasutajanime ja parooliga sisenedes veebirakendusse QR-koodi sidumiseks genereerida. MF10 nõue jäi täitmata, sest rakendus peab eelnevalt läbima ka klientide seas testimise ning seejärel ka nende tagasiside põhjal lisafunktsioone lisada ja vajadusel parandusi rakendada. Lisaks oli algselt plaanis oli ka lisada Amphora veebirakendusse vaade, kus kliendil on võimalik näha lisatud seadmeid ning neid vajadusel kustutada, kuid kahjuks jäi see töömahu tõttu saavutamata.

Väärtuslikuks osutus ka Amphora arendajate abi veebirakenduse tagasüsteemi keerukamate komponentide juures, kus heaks abiks olid olemasoleva süsteemi loogika selgitamine ja tugi serveripoolsete lahenduste integreerimiseks. Samuti oli serveri poolsete lahendustega vajalik meeskonnaga suurem kooskõlastamine kui mobiilirakendusega, sest olemasolev lahendus on pidevas arenduses ning klientide poolt igapäevaselt kasutatud, see võimaldab hiljem lihtsamalt

kanda projekti reaalajas toimivasse versiooni. Lisaks aitab kooskõlastamine kaasa hiljem rakenduse üle võtmisele.

5.3 Soovitused

Lähtudes rakenduse arendusprotsessist, võib teha mitmeid soovitusi Amphora teenuse edasiarenduseks. Tulevikus võiks dokumentide allkirjastamise ja saatmise funktsioonid olla mobiilirakenduse sisse integreeritud, et pakkuda terviklikumat kasutajakogemust. Hetkel on need funktsioonid eraldatud, mis nõuab kasutajalt veebirakenduse poole pöördumist. Integratsioon vähendaks kasutaja vajadust navigeerida allkirjastamiseks tagasi veebirakenduse juurde ja muudaks töövoo täielikult mobiilirakenduse siseseks. Turvalisuse ja mugavuse seisukohast oleks tulevikus hea lisada rakendusele ka biomeetrilise autentimise võimalused. Tänapäevased nutiseadmed toetavad sõrmejälje või näotuvastust, mida võiks kasutada täiendava turvakihina, eriti rakenduse käivitamisel.

Lisaks võiks turvalisuse seisukohalt tulevikus Amphora veebikeskkonda implementeerida täiendava kaheastmelise autentimise kihi kasutajanime ja parooliga sisselogimisel. Lahenduses saadaks server pärast kasutajanime ja parooli sisestamist automaatse teate kasutaja mobiilirakendusele. Rakendus kuvaks kasutajale kinnituspäringu, mis sisaldaks sisselogimise kellaaega ja asukohta, võimaldades kasutajal autentimiskatset kas kinnitada või tagasi lükata. Selline täiendav mehhanism pakuks turvalisuse tõstmist tavapärase paroolipõhise autentimise korral, aidates tuvastada võimalikud volitamata sisselogimiskatsed ning võimaldades kasutajal neid koheselt blokeerida.

5.4 Testimise tagasiside

Kõikide testimise tulemuste põhjal tehakse vajalikud parandused ja kohandused rakendusele. Testimisprotsess annab ühtlasi väärtuslikku tagasisidet tulevasteks arendusteks ning funktsionaalsuse laiendamiseks tulevikus. Amphora arendajate tagasiside rakendusele oli üldjoontes positiivne. Eriti tõsteti esile rakenduse kasutajaliidese lihtsat ja intuitiivset ülesehitust ning QR-koodil põhineva seostamise (autentimise) mugavust. Töövoo algatamiseks on võimalik Amphora veebirakenduses genereerida kasutajal QR-kood, mis algatab seadme sidumise protsessi. Kaamera abil suudab mobiilirakendus tuvastada QR-koodi probleemideta. Mitte ettenähtud QR-koodi lugemisel katkestab rakendus autentimise ja kuvab veateate. Autenimisprotsessi katkestades

kuvab mobiilirakendus vastavad veateated, mis on vajalik funktsioon kasutajale. Rakendus tuli testimise käigus proovitud rohkete päringutega toime hästi ning suutis tööülesanded edukalt pärida.

Edaspidise arenduse kohalt oli soovitusi palju. Esmatähtis oleks tööülesannete filtreerimise ja sorteerimise loogika täiustamine. Hetkel kuvab mobiilirakendus ülesandeid uuemast alates kronoloogiliselt, kuid ei arvesta nende prioriteetsust. Optimaalne lahendus oleks mitme välja alusel sorteerimine – esmalt prioriteedi ja seejärel kuupäeva järgi, mis tagaks, et kriitilise tähtsusega ülesanded oleksid alati kasutajale esmajärjekorras nähtavad, sõltumata nende loomise ajast. Samuti peaks olema võimalik filtreerida ülesandeid nende liigi järgi, see ühtlustaks mobiilirakenduse funktsionaalsust veebirakendusega. Näiteks Amphora veebirakenduses on tööülesanded filtreeritavad nelja kategooria abil: kooskõlastamiseks, täitmiseks, teadmiseks ja allkirjastamiseks. Lisaks on veebirakenduses erinevatele kategooriatele määratud erinevad värvid, näiteks kooskõlastamiseks on punane värv, täitmiseks sinine, teadmiseks on roheline ning allkirjastamiseks on kollane. Värvilahendused võiks olla implementeeritud ka mobiilirakendusse, et kasutajal oleks lihtsam ja intuitiivsem jätkata kasutamist.

Tagasiside käigus tuli ka välja, et mobiilirakendusel võiks olla profiilivaade, mis võimaldaks kasutajal mobiilirakenduses veenduda, millise Amphora keskkonnaga kasutaja hetkel aktiivne on. Kasutajad võivad kuuluda mitmesse Amphora keskkonda (näiteks kuuludes erinevatesse omavalitsustesse). Hetkel võimaldab mobiilirakendus siduda vaid ühe Amphora keskkonna (üks URL) ja ühe kasutajakontoga, mis seab piirangud mitmes keskkonnas tegutsevate kasutajate jaoks.

6. Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli arendada Amphora Infohaldus OÜ-le mobiilirakendus, mis võimaldab turvalist juurdepääsu Amphora infohaldussüsteemis käsitlevatele tööülesannetele, sidudes seadme QR-koodi abil. Eesmärk oli luua lahendus, mis võimaldaks kasutajatel turvaliselt ja mugavalt autentida end rakendusse ning saada ligipääs tööülesannetele.

Enne rakenduse arendamise alustamist püstitati selged funktsionaalsed ja mittefunktsionaalsed nõuded, mis hõlmasid QR-koodi abil autentimist, seadme tuvastust, andmete turvalist käsitlemist, tööülesannete ja veateadete kuvamist. Samuti teostati analüüs, testides Eestis saadaval olevatest mobiilirakendustest autentimise lahendustega nagu LHV pank, Eesti terviseportaal ja WhatsApp, et kaardistada olemasolevaid autentimislahendusi ja nende kasutajasõbralikkust.

Töö tulemusena on valminud toimiv prototüüp mobiilirakendusest, mis täitis enamiku funktsionaalseid ja mittefunktsionaalseid nõudeid. Amphora veebirakenduses on võimalik genereerida seadme lisamiseks QR-kood, mis eduka skaneerimise järel avab tööülesannete vaate mobiilirakenduses. Rakendus kuvab kasutajale tööülesanded nii tehtud kui ka tegemata, mis on REST API abil päritud veebirakendusest. Arendatud rakendus võimaldab tööülesannet indviduaalselt vaadata ehk kasutada detailvaadet. Valminud lahendus täitis peaaegu kõik seatud funktsionaalsed ja mittefunktsionaalsed nõuded. Ainsana jäid lõputöö raames täitmata kasutaja poolne seadmete haldus veebikeskkonnas ning rakenduse ametlik avaldamine Google Play ja Apple Store'is, mis on plaanis tulevikus. Töö avaldamise kuupäeval ei ole Amphora mobiilirakendus avalikult kättesaadav, sest rakendus läheb privaatsele testimisele klientidele ning seejärel kooskõlastatakse, millistel tingimustel rakendus avalikuks läheb.

Kasutatud allikad

[1] Riigi Infosüsteemi Amet. (2025). *Küberturvalisuse aastaraamat 2025*. https://www.ria.ee/sites/default/files/documents/2025-02/RIA-kuberturvalisuse-aastaraamat-2025.pdf (09.05.2025)

[2] e-Teatmik: IT ja sidetehnika seletav sõnaraamat. http://www.vallaste.ee. (30.11.2023)

[3] Digipädevus: Sõnastik. https://digipadevus.ee/sonastik/#sonastik (30.11.2023)

[4] Dhruv Bhanderi; Meet Kavathiya; Tushar Bhut; Hargeet Kaur; Meet Mehta, 2018, Impact of Two-Factor Authentication on User Convenience and Security, IEE Xplore, https://ieeexplore-ieee-org.ezproxy.utlib.ut.ee/document/10112421 (30.11.2023)

[5] Chen, C. (2017). QR Code Authentication with Embedded Message Authentication Code. *Mobile Networks and Applications*, 22, 383–394. https://doi.org/10.1007/s11036-016-0772-y (28.03.2025)

[6] Authgear. (2024). *What Is SMS Authentication and Should You Implement It?* https://www.authgear.com/post/sms-authentication-should-you-implement (09.05.2025)

[7] SK ID Solutions AS. (2024). Smart-ID. https://www.smart-id.com/et/smart-id/ (13.05.2025)

[8] SK ID Solutions AS. (2024). Mis on Mobiil-ID? https://www.mobiil-id.ee/mis-on-mobiil-id/ (13.05.2025)

[9] Riigi Infosüsteemi Amet. (2022). Riigi SSO teenus (GovSSO). https://e-gov.github.io/GOVSSO/ (28.03.2025)

[10] SK ID Solutions. (2022). Hinnakiri. https://www.skidsolutions.eu/price-list/ (28.03.2025)

[11] SK ID Solutions. Price list. (2024). https://www.skidsolutions.eu/price-list/#mobile-id (06.04.2025)

[12] Google. (2024). Two-factor authentication can help users to better protect themselves online. Google Account offers several options. https://safety.google/stories/passwort/ (14.05.2025)

[13] Amphora Infohaldus OÜ. (2021). https://www.amphora.ee/default.aspx?menu=3302&loc=03 (05.04.2025)

[14] Microsoft. (2024). C# Version History. https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history (05.04.2025)

[15] Statista. Most used programming languages among developers worldwide as of 2024. https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/

[16] Brainhub. (2023). .NET MAUI in a Nutshell. https://brainhub.eu/library/net-maui-in-nutshell (05.04.2025)
[17] Microsoft. (2024). What is .NET MAUI. https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui?view=net-maui-9.0 (05.04.2025)
[18] What is a REST API? IBM. https://www.ibm.com/topics/rest-apis#:~:text=the%20next%20step-,What%20is%20a%20REST%20API%3F,representational%20state%20transfer%20architectural%20style. (02.12.2023)
[19] MDN Web Docs. An overview of HTTP. https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/Overview (15.04.2025)
[20] QR Code.com. (n.d.). *What is a QR Code?* https://www.qrcode.com/en/about/ (16.04.2025)
[21] Tan Jin Soon. (2008). *Three QR Code*. Synthesis Journal. https://doi.org/10.5120/three_qr_code (16.04.2025)
[22] WhatsApp. (2024). WhatsApp Web – How it works. https://www.whatsapp.com/faq (06.04.2025)
[23]David C. (2024, 8. veebruar). *QR Codes – what's the real risk?* National Cyber Security Centre. https://www.ncsc.gov.uk/blog-post/qr-codes-whats-real-risk
[24] Hasura. (2024). *What is MSSQL Server?* https://hasura.io/learn/database/microsoft-sql-server/what-is-mssql/ (06.05.2025)
[25] Microsoft Learn. Model-View-ViewModel - .NET. https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm (15.04.2025)
(15.04.2025)
[26] Chitrasingla2001. Functional vs Non Functional Requirements. Geeks For Geeks, 02.12.2022. https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/ (02.12.2023)
[27] Codacy. (2023). *What is clean code?* Kättesaadav aadressil: https://blog.codacy.com/what-is-clean-code (05.05.2025)
[28] LHV. Mobiiliäpp. https://www.lhv.ee/et/mobiiliapp (03.01.2024)
[29] Terviseportaal. https://www.terviseportaal.ee/ (05.01.2024)
[30] Chao Shen. User practice in password security: An empirical study of real-life passwords in

the wild? ScienceDirect, 08.2016. https://www.sciencedirect.com/science/article/abs/pii/S0167404816300657 (05.01.2024)

Lisad

I. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Ron-Aran Paju

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose "Amphora Infohaldus OÜ rakenduse ja kaheastmelise autentimise arendamine", mille juhendaja on Kristiina Rahkema,

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada Tartu Ülikooli digitaalarhiivi kuni autoriõiguse kehtivuse lõppemiseni;

2. annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi kaudu Creative Commonsi litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni;

3. olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile;

4. kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Ron-Aran Paju

15.05.2025