TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Informaatikainstituut

IAG40LT

Tristan Tomilin 135065IAPB

EESTI ID-KAARDI JA MOBIIL-ID TOIMINGUTE TEAVITAMISRAKENDUS ANDROID PLATVORMIL

Bakalaureusetöö

Juhendaja: Paul Leis
PhD
Dotsent

Tallinn 2016

Autorideklaratsioon

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

Autor: Tristan Tomilin

16.05.2016

Annotatsioon

Käesoleva bakalaureusetöö ülesandeks on luua ülevaade mobiilirakendusest, mis teavitab isikut tema ID-kaardi ja Mobiil-ID digitaalsest kasutusest. Rakenduse otstarve on maandada digitaalse identiteedi kasutusega kaasnevaid turvaohte ning leevendada nende tagajärgi. Projekt valmib ürituse EV100 raames. Töös on loodud ülevaade mobiilirakendusest, kirjeldatud selle arhitektuuri ja kasutajaliidest ning toodud välja võimalikud kasutusjuhud. Töö annab ülevaate rakenduse ja serveri andmevahetuseks rakendatud teenustest, veebiserveri suhtlusest andmebaasiga ning veebirakenduse abil kasutaja mobiilseadmete haldamise protseduuridest. Isiku digitaalsete toimingute pärimisteenust pakub AS Sertifitseerimiskeskus, kes ühtlasi hakkab haldab veebiserverit ja andmebaasi.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 40 leheküljel, 3 peatükki, 26 joonist, 3 tabelit.

Abstract

Estonian ID-card and Mobile-ID usage notifier application for Android

The aim of this bachelor's thesis is to provide an overview of an application that notifies the user if his/her ID-card or Mobile-ID has been digitally used. The purpose of the application is to alleviate the consequences of the security risks that occur regarding digital identity services. The project will be carried out in connection with the 100th national birthday of Republic of Estonia.

The thesis provides an overview of the mobile application, describing its layered architecture and use cases in detail. The user interface is built in accordance with the principles of material design. The main functionalities of the mobile application include viewing, sorting and filtering digital identity action logs and configuring notification settings. Mobile devices need an active session with the server to request the user's digital identity action logs. A web application is used for registering and removing mobile devices. The user needs to provide authentication to access the web application.

The communication between mobile applications and the web server is defined through an API that uses the HTTP protocol. The services follow REST architectural style. The web server communicates with the database using PL/SQL. The service for retrieving a person's digital identity action logs is provided by Certification Centre Ltd. who is also responsible for maintaining the web server and the database.

Retrofit, JSON Web Token, QR-code and X-road are the main used technologies regarding the project. Each serves a specific purpose and benefits to the usage and workflow of the application.

The thesis is in Estonian and contains 40 pages of text, 3 chapters, 26 figures, 3 tables.

Lühendite ja mõistete sõnastik

Android	Tarkvarakomplekt elektroonikaseadmetele, mis hõlmab operatsioonisüsteemi, vahetarkvara ja peamisi rakendusi
API	*Application Programming Interface*, rakenduse programmeerimise liides
APK	*Android Application Package*, androidi rakenduse pakett
ASCII	Ameerika Informatsioonivahetuse Standardkood
Base64	Kodeering, mis esitab binaarkujul andmeid *ASCII* formaadis
GET	Hüperteksti edastusprotokolli päringu meetod, mis on mõeldud andmete pärimiseks
EV100	Eesti Vabariigi 100. sünnipäeva raames korraldatud üritus
Fragment	Representeerib Android rakenduse käitumist või kasutajaliidese mingit osa
HTTP	*Hypertext Transfer Protocol*, hüperteksti edastusprotokoll – teabe edastamiseks loodud protokoll arvutivõrkudes
hüpertekst	Hüperlinkidega tekst (näiteks veebileht)
IDE	*Integrated Development Environment*, integreeritud arenduskeskkond
JavaScript	Interaktiivsete veebisaitide loomiseks mõeldud programmeerimiskeel
JSON	*JavaScript Object Notation*, lihtsustatud andmevahetusvorming, mis põhineb *JavaScripti* programmeerimiskeele alamhulgal
JWT	*JSON Web Token*, standard, mis kirjeldab kompaktset ja iseseisvat meetodit osapoolte vahelise informatsiooni turvalist vahendamist
Link	Pikemalt hüperlink – hüpertekstis viide teisele dokumendile, selle osale või sama dokumendi muule osale
Ltd	*Limited company*, aktsiaselts
Material design	*Google*'i arendatud disainimisprintsiipide kogum
Metaandmed	Andmeid kirjeldav informatsioon
MVC	*Model-View-Controller*, kasutajaliideste loomiseks mõeldud tarkvaraarhitektuuri muster
Offline	Võrgust väljas

Oracle	Objekt-relatsiooniline andmebaasi haldussüsteem
payload	Sõnumi tegelik sisu, kuhu ei kuulu päis ega metaandmed
POST	Hüperteksti edastusprotokolli päringu meetod, mis on mõeldud andmete saatmiseks
Primaarvõti	Kandidaatvõti, mis on valitud relatsiooni kirjeid unikaalselt identifitseerima
QR-Kood	*Quick Response Code*, kahemõõtmeline maatrikskood, mis võimaldab skaneerida infot mobiiltelefoni
REST	*Representational State Transfer*, tarkvaraarhitektuuri stiil, mis on kasutusel hüpermeedia hajussüsteemide valdkonnas
Retrofit	Tüübiturvaline hüperteksti edastusprotokolli raamistik Android arendusplatvormile
SDK	*Software Development Kit*, kogumik tarkvaraarenduse tööriistu, mis võimaldavad kindla tarkvara paketi loomist
Service Unavailable	*HTTP* veakood 503. Teenus ei ole kättesaadav.
Sessioon	Interaktiivne järjestikune informatsiooni vastastikune vahetus
SHA-256	*Secure Hash Algorithm*, krüpteerimisalgoritm, mis kasutab 32 bitist kodeeringut
Sool	Juhuslik sõne, mis on sisendiks räsifunktsioonile
SQL	*Structured Query Language*, andmebaasi päringukeel
Sõne	Sõnaeksemplar, kirjatähtede jada või kõneühik
Token	Unikaalne identifikaator (tavaliselt räsi kujul), mis identifitseerib serveri ja kliendi vahelisi sõnumeid
Unauthorized	*HTTP* veakood 401. Volituseta.
Utiliit	Spetsiifilist ülesannet täitev programm
x-header	Ebastandardne *HTTP* päringu päise parameeter
X-tee	Tehniline ja organisatsiooniline infosüsteemide andmevahetuskiht, mis võimaldab registritel ja andmekogudel omavahel turvaliselt ning teatud volituste piires suhelda
XML	*Extensive Markup Language*, standardne üldotstarbeline märgistuskeel info edastuseks

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Käesoleva bakalaureuse töö eesmärgiks on luua kirjeldus mobiilirakendusest, et täiendavalt tagada digiteenuste turvalisust Eesti Vabariigi kodanikele. Rakendus teavitab isikut tema ID-kaardi ja Mobiil-ID digitaalsest kasutusest. Personaalsed turvaandmed võivad sattuda valedesse kätesse ning isiku identiteeti võidakse kuritarvitada. Elektroonilisel isikutuvastusel ja allkirjastamisel on oht, et teenust kasutab võõras isik, identiteedi omaniku enese teadmata. Pakutav lahendus teavitamisrakendusest maandaks turvaohte ning leevendaks võimalikke tagajärgi.

Projekti läbi viimist on arutatud Riigi Infosüsteemi Ameti arhitekti Andres Kütiga, kes kiitis idee heaks. Samuti on projekt kooskõlastatud Sertifitseerimiskeskuse tootejuhiga Urmo Keskel, kes andis nõusoleku, et projektiga seotud veebirakendust, -serverit ja andmebaasi hakkab haldama AS Sertifitseerimiskeskus [1] . Veebiserver pärib isiku digitaalsete toimingute andmeid teenuselt, mida hakkab pakkuma samuti Sertifitseerimiskeskus. Projekt teostatakse ürituse EV100 raames.

Mobiilirakendus kasutab veebiserverit sessiooni aktiveerimiseks, isiku digitaalsete toimingute pärimiseks, sessiooni olemasolu kontrollimiseks ning sessiooni tühistamiseks. Andmevahetus toimub *REST* [2] metoodika printsiipide põhjal, kasutades *HTTP* päringuid. Veebirakendus on eraldiseisev keskkond, kus isik saab toimingute teavitamisrakendust kasutavaid mobiilseadmeid hallata.

Töö tulemusena valmib mobiilirakenduse ja sellega seotud komponentide kirjeldus. Rakendus teavitab isikut, kui on kasutatud tema ID-kaarti või Mobiil-ID'd digitaalsete toimingute teostamiseks. Kasutajale kuvatakse rakenduses nimekiri tema toimingutest. Igal kirjel on märgitud aeg, kasutatud sertifikaadi staatus, teenust osutav kanal, toimingu tüüp ning kasutatud vahend. Tuntud kanalite (eesti.ee, emtak.riik.ee, politsei.ee, swedbank.ee, seb.ee, jm) puhul on teenusepakkujad tähistatud ettevõtte logoga. Toimingute nimekirja on võimalik parameetrite põhjal sorteerida ja filtreerida.

Töö põhiosa on jaotatud kolmeks peatükiks. Teoreetilises osas on kirjeldatud digitaalsete toimingute kaasnevaid turvaohte ja projektis kasutatud tehnoloogiaid. Veebiserveri alamjaotuses on kirjeldatud andmemudelit ja andmebaasiga suhtlust, veebirakenduses mobiilseadmete haldamist ning teenuseid, mis on kasutusel serveri mobiilirakenduse andmevahetuseks. Töö põhiosa viimases osas on toodud ülevaade mobiilirakenduse kasutusjuhtudest, kasutajaliidesest ja arhitektuurist. Töö praktiliseks väljundiks on Android rakenduse prototüüp, mis ei kasuta toimivat serverit ega andmebaasi.

2 Teoreetiline osa

2.1 Turvalisus

Eestis on digitaalsel identiteedil kujunenud väga laialdane kasutusvaldkond. Üha enam luuakse uusi teenuseid, mis on digitaalselt kättesaadavad. Mitmed toimingud ja tehingud, mis varem nõudsid füüsiliselt isiku kohalolu ning paberkandjat, on nüüdseks teostatavad veebi kaudu, vajades digitaalseid sertifikaate ja vastavaid paroole. Enamlevinud kasutusvaldkonnad on e-pangandus, isikutuvastus veebiportaalides, e-hääletamine ning muud riigiportaali e-teenused [3] , [4] .

Digiteenuste valdkonna laienemine süvendab ühtlasi ka privaatsuse ja turvariske. Küberkuritegevus ning petuskeemid on tõusvas joones, sest arvutisüsteem on üks soodsaimaid keskkondi kuritegevuseks, kuna toimepaneku faktist ei pruugi mingit jälge jääda [5] . Välja petetud isikuandmed võivad kaasa tuua identiteedivarguse. Isiku digitaalse identiteediga võidakse tema enese teadmata osta kaupu, tarbida teenuseid ning panna korda kuritegusid [6] . Üks enamlevinud petuskeemi meetod on *õngitsemine*, mille käigus antakse kasutajale mingil ettekäändel link veebilehele, mis on pealtnäha sama välimusega, mis panga või muu asutuse oma [7] . Kui veebisait kasutajale ootuspäraselt käitub, ei teki kahtlust, et paroolide sisestamine endast mingit ohtu kujutab. Nii võib kasutaja alles tükk aega hiljem avastada, et tema identiteeti on kuritegelikult kasutatud või on tema pangakonto säästud märgatavalt kahanenud. *Õngitsemine* on ühtlasi üks peamisi ID-kaardi ja Mobiil-ID paroolide nuhkimiseks kasutatavaid meetodeid. Teine oluline turvaoht on erinevat tüüpi pahavara, mis kasutaja mobiiltelefoni või isiklikku arvutisse sattudes salvestab ja saadab kuritegijaile paroole.

Eelpool mainitud turvaohtude tagajärgi leevendab asjaolu, et isik saab teavituse, kui on kasutatud tema digitaalselt identiteeti. Mida varem isik laseb oma isikutuvastuse ja digiallkirjastamise sertifikaadid peatada, seda väiksem on potentsiaalne tekitatud kahju. Samuti suurendab see võimalust kurjategijaile jälile saamiseks.

2.2 Kasutatud tehnoloogiad

Mobiilirakenduse ning sellega seonduvate komponentide tarbeks on kasutusel tehnoloogiad, mille hulgast ülevaade on toodud järgmistest: *JWT*, *Retrofit*, *QR*-kood ning X-tee.

2.2.1 JSON Web Token

Rakenduse ja serveri vahelise suhtluse vahendamiseks on kasutusel *JWT*. Selles sisaldub päis, *payload* (Joonis 1) ning signatuur, mis tõendab saabuva info usaldusväärsust.

```
{
    "certificate_id": "2QPH49TH65RAw4WZo4mGiYOBo4VA"
    "session_id": "1234657890"
    "device_id": "f07a13984f6d116a"
    "country_code": "EST"
    "registration_code": "47101010033",
    "expiration_date": "2016-04-23T18:25:43"
}
```

Joonis 1. Näide: *JWT payload*.

Allkirjastamine toimub avaliku ja privaatvõtme paari abil kasutades *SHA-256*. *JWT* tehnoloogia kasutamise eelised on kompaktsus ja sõltumatus. Kompaktsus väljendub *tokeni* minimaalses andmemahus. Väike maht tagab omakorda parema edastuskiiruse. Sõltumatus väljendub vajalike kasutaja andmete olemasolus, mistõttu ei ole tarvis andmebaasist lisa päringuid teha.

Sessiooni aktiveerimise päringul koostab server *JWT* rakenduselt saadud ja andmebaasist päritud andmete põhjal. Server lisab *tokeni HTTP* päringu vastuse sisuna ja saadab tagasi. Kuna *token* sisaldab konfidentsiaalselt infot, krüpteerib rakendus selle, kasutades seadme unikaalset identifikaatorit ja soola [12] . Rakendus salvestab tulemuse lokaalselt seadme sisemällu. Krüpteerimine tagab *tokeni* sisu loetamatuse väljaspool toimingute teavitamisrakendust. Kogu protsess loob kestva seisundivaba sessiooni serveri ja rakenduse vahel. Iga järgneva rakenduse päringu korral kontrollib server, kas saabuva sõnumi päis sisaldab korrektse sisuga *tokenit*. Vastasel juhul tagastab server vea. Käesolev mobiilirakendus edastab *JWT* serverile *HTTP* päises.

JWT kehtivus on piiratud kasutaja autentimisvahendi sertifikaadi kehtivusaja lõpuga. Kui sertifikaat kaotab kehtivuse, tühistab server sessiooni. Rakenduse edasiseks kasutamiseks antud seadmes isikul ennast autentida kehtiva sertifikaadiga, mille korral luuakse uus *JWT*.

2.2.2 Retrofit

Serveri ja rakenduse vahelise suhtluse puhul on oluline seisundi vabadus. Suhtluse vahendamisel on kasutusel *Retrofit* raamistik, mis põhineb *REST* tarkvaraarhitektuuril [10] .

REST metoodika kasutamise põhjused antud kontekstis on järgmised:

- Kasutaja toiminguid ei saa pärida tihenimi kui üks minut. Lisaks võivad rakendused töötada mobiilseadmete taustal ööpäevaringselt. Seetõttu ei ole otstarbekas hoida mälus iga sessiooni seisundit.

- Iga päring on sõltumatu ja lõplik. Server identifitseerib rakenduselt tulnud päringu päises asuva *tokeni* põhjal. Seetõttu puudub vajadus hoida andmeid sessiooni seisundi kohta, mis hoiab mäluruumi kokku.

- Vahemälu kasutus vähedab päringu andmemahtu. Rakendus pärib ainult uusi kasutaja toiminguid, vanad salvestatakse vahemällu.

- Mobiilirakendusel ja serveril on ühine arusaam teenuste kaudu vahendatavatest andmetest. Liidest ei ole *REST* teenuste puul võimalik reglementeerida.

Toimingute pärimise teenus on muudetud *Retrofiti* abil Java liideseks. Iga teenuse väljakutse muudetakse asünkroonseks *HTTP* päringuks veebiserverile ning pöördutakse määratud aadressi poole. Näidiskood kasutusest on toodud järgneval joonisel (Joonis 2).

```
public interface DigitalIdentityService {

    @GET("api/identity/log?date_from={timestamp}")
    Call<List<Action>> getActions(@Path("timestamp") String timestamp);
}

Retrofit retrofit = new Retrofit.Builder()
    .baseUrl("https://sk.ee/{api}")
    .build();

DigitalIDentityService service = retrofit.create(DigitalIDentityService.class);

Call<List<Action>> actions = service.getActions("2016-05-17T11:16:05");
```

Joonis 2. Näide: *Retrofit API*.

2.2.3 QR-kood

QR-kood on kodeeritud kahemõõtmeline maatrikskood, mis võimaldab skaneerida infot mobiiltelefoni, kus mobiilrakendus selle dekodeerib [20] . Sessiooni aktiveerimisprotsessi käigus skaneerib kasutaja talle kuvatud *QR*-koodi, misjärel skaneerimisrakendus dekodeerib selle. Tulemuseks on aadress, mis sisaldab viidet toimingute teavitamisrakenduseni ning aktiveerib sama protsessi, mis aktiveerimiskoodi käsitsi sisestamine. Aadress sisaldab parameetrina ka aktiveerimiskoodi ennast.

Tehnoloogia on kasutusel, et hõlbustada kasutajale autentimisprotsessi rakenduse kasutamiseks mobiilseadme lisamisel. Aktiveerimiskoodi käsitsi sisestamisel on suur tõenäosus, et kasutaja eksib, kuna ekraanil kuvatud kood on pikk ja juhuslik. *QR*-koodi skaneerimine automatiseerib sisestamisprotsessi. Meetodi puuduseks on asjaolu, et kasutajal võib puududa mobiilseadmest skaneerimisrakendus või oskus selle opereerimiseks.

2.2.4 X-tee

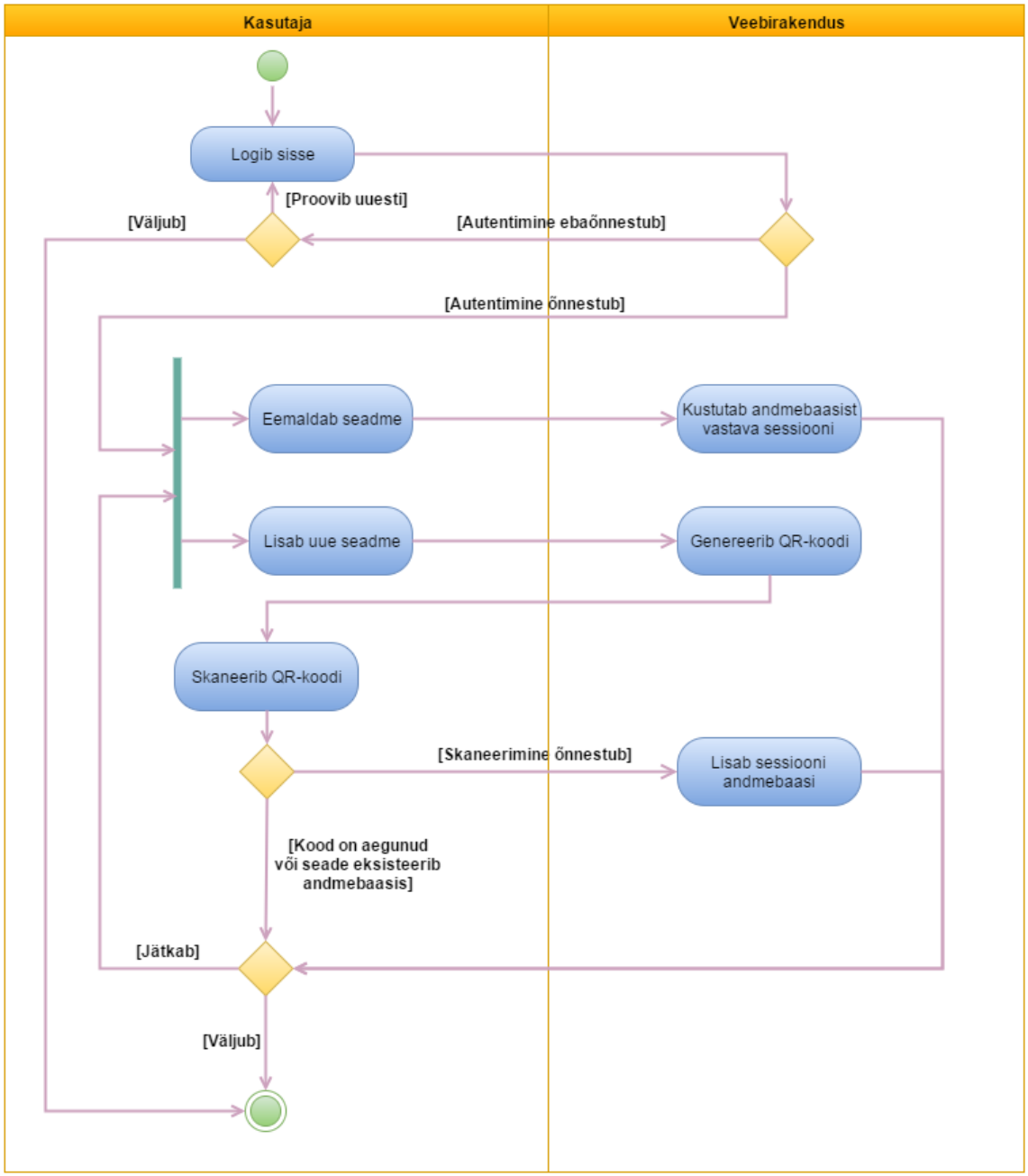
X-tee on turvaline, hajutatud teenussiin organisatsioonide vahelise andmevahetuse korraldamiseks. Informatsiooni vahendamisel hoolitseb X-tee andmete krüpteerimise eest ning lisab edastatavatele sõnumitele autentsust tõendava digitaalallkirja. Tänu allkirjale saavad osapooled olla kindlad vahetatavate andmete autentsuses [21] . Infoedastus X-tee kaudu toimub eelnevalt defineeritud kasutusmallide ning andmeteenuste piires. Andmete vorming on üheselt määratud [9] .

Isiku digitaalse identiteedi toimingute pärimisteenuse liitmine X-teega võimaldaks turvalisemat ja stabiilsemat andmevahetust veebiserveri ja Sertifitseerimiskeskuse poolt pakutava teenuse vahel. Antud kontekstis ei ole tegemist delikaatsete isikuandmete käideldavusega [8] , kuid sellest hoolimata on turvalisuse aspekt väga oluline. Lisaks saab veebiserver X-tee kaudu kasutada isikuandmete volituste teenust. Selle abil oleks rakenduses kuvatud ka isiku antud volitustega seotud toiminguid.

3 Veebiserver

3.1 Seadmete haldus

Toimingute teavitamisrakendus on paralleelselt kasutatav mitmes mobiilseadmes. Seadmete lisamine ja eemaldamine on võimalik vaid veebirakenduses. Veebirakendusse sisenemiseks tuleb kasutajal end turvalise vahendiga (ID-Kaart, Mobiil-ID) autentida. Eduka autentimise korral kuvatakse kasutajale seadmete haldamise vaade, kus on nimekirjana esitatud kõik registreeritud seadmed. Seadmeid saab lisada *QR*-koodi skaneerimise ja aktiveerimiskoodi käsitsi sisestamise abil. Aktiivse sessiooni lõpetamiseks tuleb kasutajal pärida vastava seadme eemaldamine nimekirjast. Kasutaja ja veebirakenduse tegevused mobiilseadmete haldamisel *QR*-koodi abil on toodud järgneval joonisel (Joonis 3).

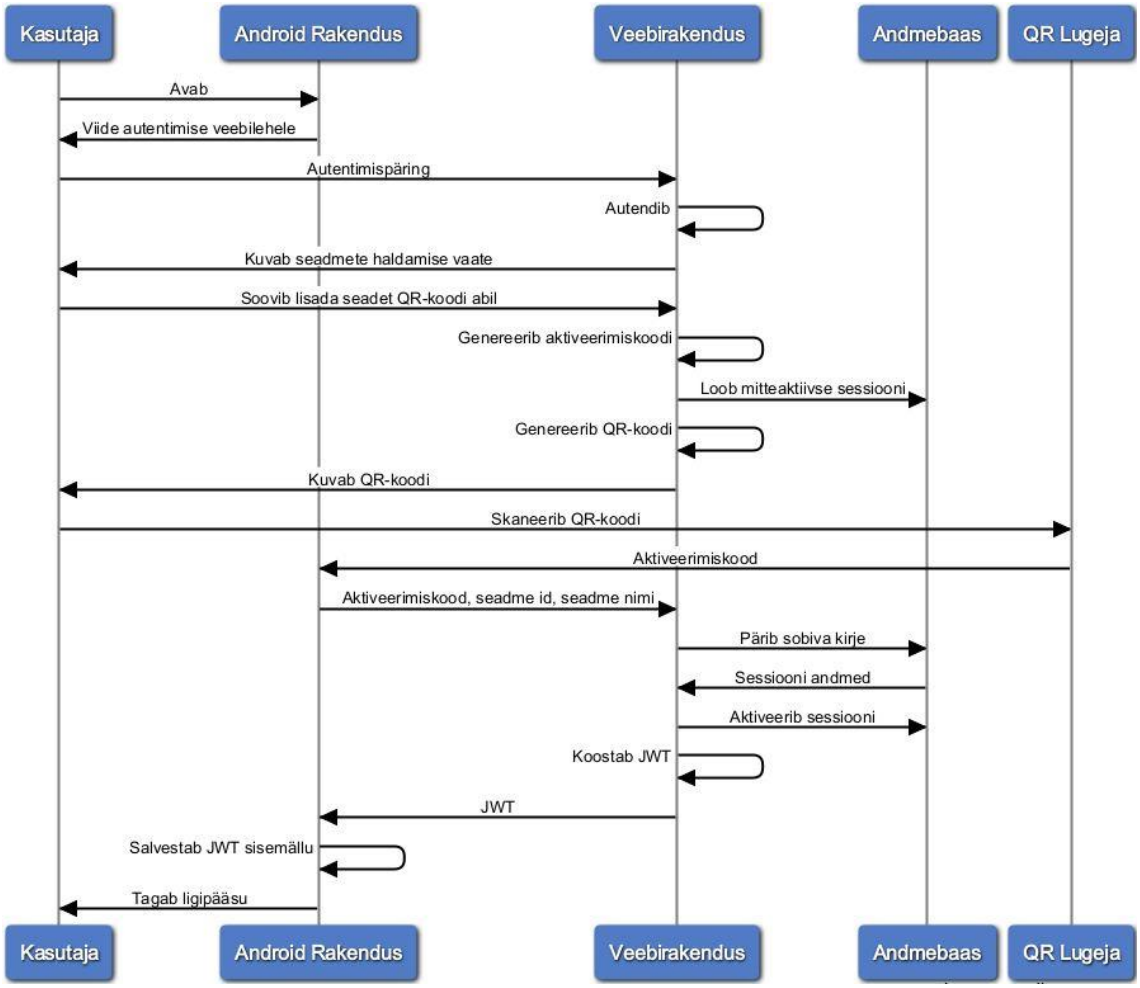


Joonis 3. Seadmete haldus *QR*-koodi näitel.

3.1.1 Seadme lisamine QR-koodi abil

Veebirakendusse õnnestunud autentimise järel valib kasutaja *QR*-koodiga seadme lisamise operatsiooni. Veebirakendus genereerib aktiveerimiskoodi ning lisab andmebaasi mitteaktiivse sessiooni kirje, milles on määratud kood, autenditud kasutaja andmed ning aegumise ajamärgend. Veebirakendus genereerib *QR*-koodi ning kuvab selle kasutajale. Kood on tarvis kahe minuti jooksul skaneerida sama nutiseadmega, millega on soov hakata kasutama teavitamisrakendust. *QR*-kood kaotab kehtivuse, kui seda on juba skaneeritud või selle loomise hetkest on möödunud kaks minutit.

Skaneerimise järel edastab *QR*-koodi lugeja teavitamisrakendusele aktiveerimiskoodi parameetrina aadressis. Rakendus saadab sessiooni aktiveerimispäringu veebiserverile koos seadme nime-, identifikaatori- ja saadud koodiga. Veebirakendus pärib andmebaasist aktiveerimata sessiooni andmed rakenduselt saadud aktiveerimiskoodi põhjal. Seejärel aktiveerib veebirakendus sessiooni, uuendades andmebaasis vastavat kirjet ning koostab saadud andmete põhjal *JWT*. Server saadab selle krüpteeritud kujul Android rakendusele, mis salvestatab selle võtme-väärtuse paarina sisemällu. Võtmena kasutatakse mobiilseadme unikaalset identifikaatorit. Salvestamiseks on kasutusel *SharedPreferences* [13] . Selle meetodika korral pääseb ilma juurõigusteta andmetele ligi ainult andmed salvestanud rakendus ise. Protsessi tulemusena on kasutajale tagatud ligipääs toimingute kuvamiseks (Joonis 4).

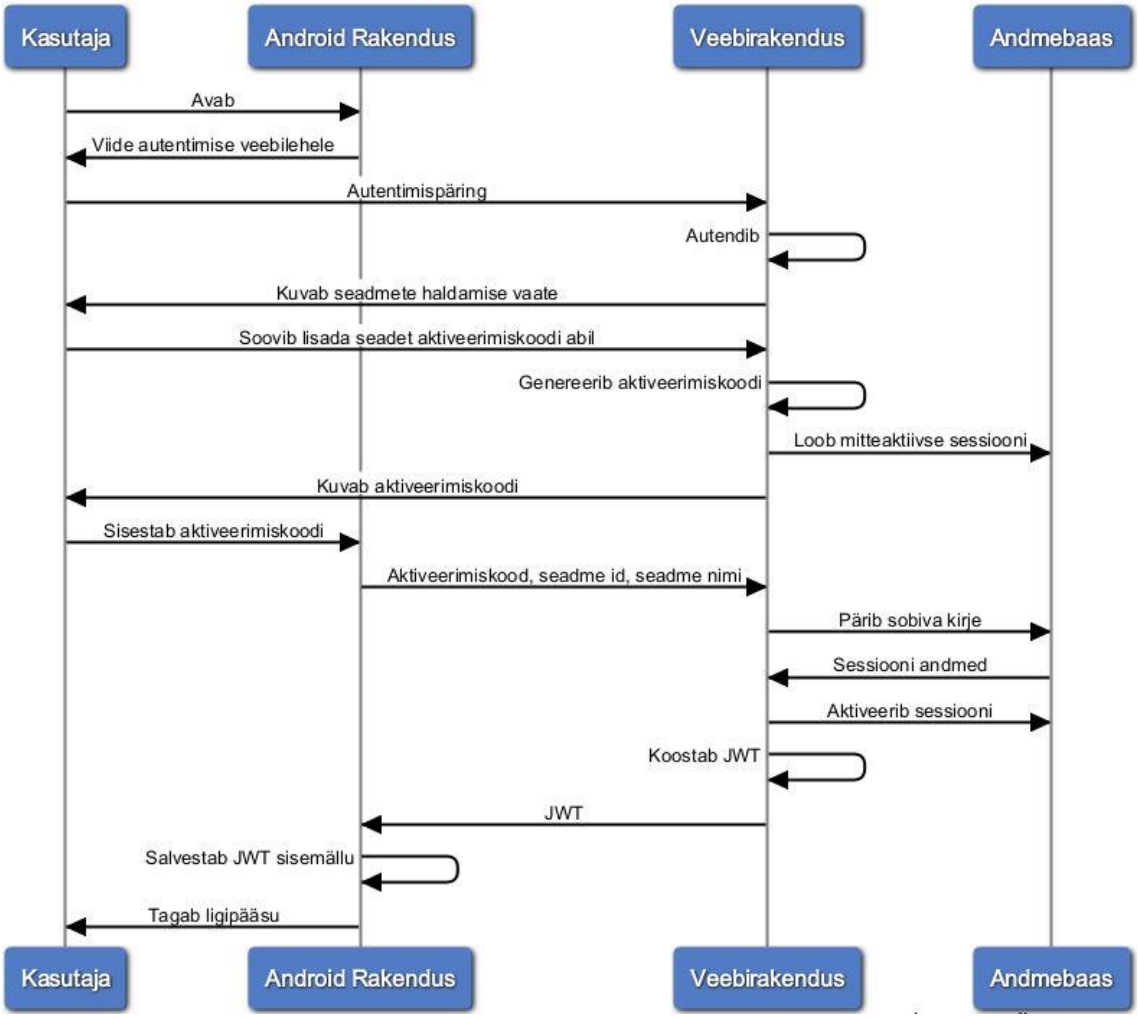


Joonis 4. Autentimine *QR*-koodi abil

3.1.2 Seadme lisamine aktiveerimiskoodi abil

Veebirakenduses autentimise järel valib kasutaja aktiveerimiskoodiga seadme lisamise operatsiooni. Veebirakendus genereerib aktiveerimiskoodi ning lisab andmebaasi

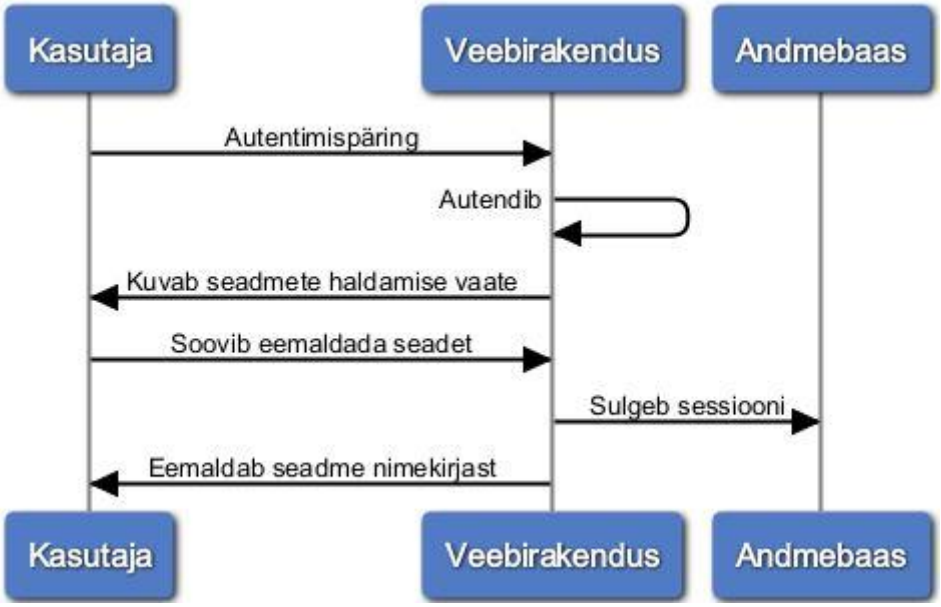
mitteaktiivse sessiooni kirje, milles on määratud kood, autenditud kasutaja andmed ning aegumise ajamärgend. Aktiveerimiskood kuvatakse kasutajale, kes peab selle käsitsi Android rakendusse sisestama. Rakendus saadab sessiooni aktiveerimispäringu veebiserverile koos seadme nime-, identifikaatori- ja kasutaja sisestatud koodiga. Edasine protsess on sama, mis *QR*-koodi abil seadme lisamise puhul (Joonis 5).



Joonis 5. Autentimine aktiveerimiskoodi abil.

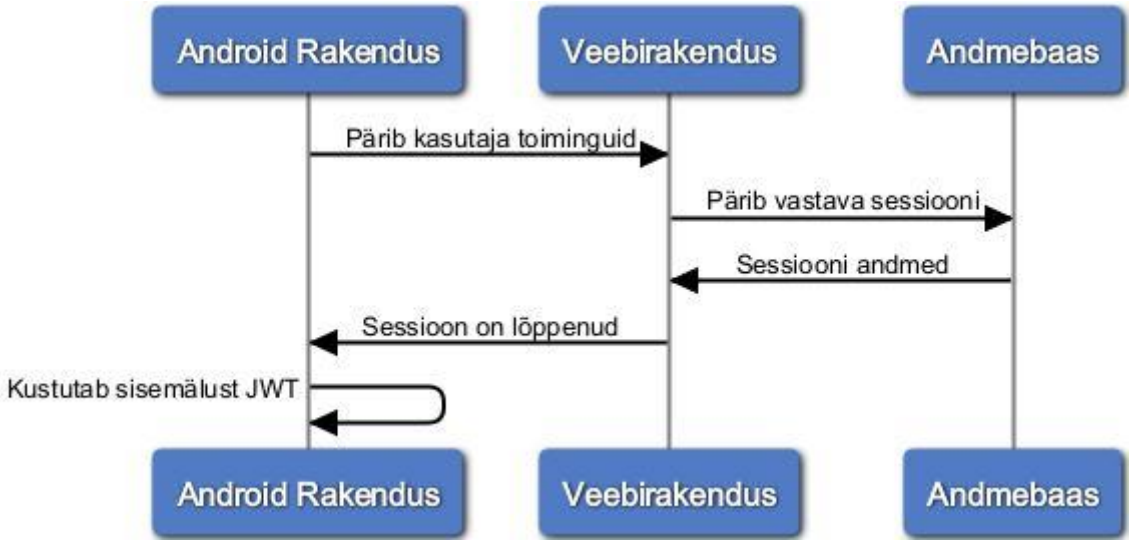
3.1.3 Seadme eemaldamine

Veebirakenduses autentimise järel valib kasutaja seadme eemaldamise operatsiooni. Veebirakendus sulgeb sessiooni, määrates andmebaasis antud sessiooni staatuseks „*revoked*“. Seejärel eemaldatakse seade haldamise nimekirjast (Joonis 6).



Joonis 6. Seadme eemaldamine

Kui töötav taustrakendus uue toimingute päringu teeb, vastab server veakoodiga 401 (*Unauthorized)* [14] , sest sessioon on tühistatud. Rakendus kustutab sisemälust *JWT* (Joonis 7) ning tühjendab vahemälust kasutaja toimingud. Kasutajale kuvatakse teade sessiooni tühistamise kohta. Kasutajal ei ole võimalik toiminguid kuvada enne mobiilseadme uuesti registreerimist.



Joonis 7. Sessiooni lõpp

3.2 Teenused

Mobiilirakendus suhtleb serveriga *HTTP* päringute abil, lähtudes *REST* metoodika printsiipidest. Serverile suunatud teenuse korral, mis nõuab seisundivabat rakenduse

autentimist, peab rakenduselt saabuv päring sisaldama päises parameetrit *Auhorization*. Väärtus on kujul „Bearer <*token*>“, kus <*token*> on krüpteeritud kujul *JWT*. Server dekrüpteerib saadud *tokeni* sisu. Esmalt autendib server päringu saatnud seadme, kontrollides *JWT* sisust loetud mobiilseadme identifikaatori vastavust *HTTP* päisesse lisatud väärtusele. Seejärel kontrollib server, kas *token* on aegunud, võrreldes sisust loetud atribuudi *expiration_date* väärtust hetkese kuupäeva ja kellaajaga. Kui *JWT* on kehtiv, pöördub server andmebaasi poole, et kontrollida sessiooni identifikaatori põhjal, kas antud sessioon on aktiivne.

3.2.1 Sessiooni aktiveerimine

Sessiooni aktiveerimiseks saadab rakendus serverile *HTTP POST* päringu, mis on toodud joonisel (Joonis 8). Päringu päisesse lisab rakendus aktiveerimiskoodi, seadme nime ja unikaalse identifikaatori.

```
POST /api/auth/activate
```

Joonis 8. Sessiooni aktiveerimise päring.

Sessiooni aktiveerimise *HTTP* päringu päise parameetrid on kirjeldatud järgnevas tabelis (Tabel 1).

Tabel 1. Sessiooni aktiveerimise päringu päise parameetrid.

Parameetri nimi	**Andmetüüp**	**Kirjeldus**	**Kohustuslik**
device_id	Sõne	Mobiilseadme unikaalne identifikaator	Jah
device_name	Sõne	Mobiilseadme nimi	Jah
activation_code	Sõne	Serveri genereeritud kood sessiooni aktiveerimiseks	Jah

Vastuvõetud aktiveerimiskoodi põhjal teeb server andmebaasi päringu, et leida vastav aktiveerimata sessioon. Kirje leidmisel andmebaasist, vastab server *HTTP* staatuse koodiga 200 [14] . Päringu vastuse sisus on *JSON* kujul atribuudi *token* väärtusena krüpteeritud *JWT* ning *tokeni* aegumise ajamärgend atribuudi *expiration_date* väärtusena (Joonis 9).

```
HTTP\1.1 200 OK
{
    token: {JWT},
    expiration_date: {timestamp}
}
```

Joonis 9. Õnnestunud sessiooni aktiveerimise päringu vastus.

3.2.2 Sessiooni kehtivuse kontroll

Kasutaja sisenemisel rakendusse, tehakse serverile *HTTP GET* päring (Joonis 10).

```
GET /api/auth/self
```

Joonis 10. Sessiooni kehtivuse kontrollpäring.

Teenuse toimimiseks peab serverile tuvastama rakenduse autentsust. Päringu seisundivaba autentimise päise parameetrid on kirjeldatud järgnevas tabelis (Tabel 2).

Tabel 2. Seisundivaba autentimist nõudva päringu päise parameetrid.

Parameetri nimi	**Andmetüüp**	**Kirjeldus**	**Kohustuslik**
Authorization	Sõne	Seisundivabaks autentimiseks lisatud parameeter kujul „Bearer <*token*>“	Jah
x-avantgo-device-id	Sõne	Mobiilseadme unikaalne identifikaator lisatud *x-header* parameetrina.	Jah
x-h3g-device-name	Sõne	Mobiilseadme nimi lisatud *x-header* parameetrina.	Jah

Päringu õnnestumise korral saadetav *HTTP* vastus on järgneval joonisel (Joonis 11). Server lisab vastuse sisusse *JSON* kujul atribuudid *status* ja *expiration_date*, mis tähistavad vastavalt sessiooni staatust ning sessiooni aegumise ajamärgendit.

```
HTTP\1.1 200 OK
{
    status: {status},
    expiration_date: {timestamp}
}
```

Joonis 11. Õnnestunud sessiooni kehtivuse kontrollpäringu vastus.

3.2.3 Viimaste toimingute pärimine

Kasutaja sätestatud teavitamise ajaintervalli möödudes, pärib rakendus serverilt isiku viimaseid toiminguid teenuse abil, mis on toodud joonisel (Joonis 12). Päring sisaldab parameetrit nimega *date_form*, mille väärtuseks on ajamärgend, millest alates uusi toiminguid päritakse. Päisesse lisatud parameetrid on kirjeldatud tabelis (Tabel 2).

```
GET /api/identity/log?date_from={timestamp}
```

Joonis 12. Viimaste isiku toimingute päring.

Õnnestunud päringu korral lisab server päringu vastuse sisusse *JSON* kujul massiivi toimingu objektidega (Joonis 13).

```
HTTP\1.1 200 OK
{
    actions: [
        {
            status: {status},
            type: {type},
            method: {method},
            service: {service},
            date: {date}
        }
    ]
}
```

Joonis 13. Õnnestunud toimingute päringu vastus.

3.2.4 Välja logimine

Kasutaja välja logimisel teeb mobiilirakendus serverile päringu, mis on toodud järgneval joonisel (Joonis 14). Päisesse lisatud parameetrid on kirjeldatud tabelis (Tabel 2).

```
GET /api/auth/logout
```

Joonis 14. Kasutaja välja logimise päring.

Sessiooni tühistamise õnnestumise korral vastab server *HTTP* staatuse koodiga 200 [14]
.

3.3 Andmebaas

Andmebaasi haldussüsteemina on kasutusel *Oracle*. Andmemudel koosneb ühest tabelist nimega „*Session*“ (Joonis 15). Tabelis hoitakse kirjeid sessioonidest.

Joonis 15. Andmemudel

Atribuutide kirjeldused on toodud järgnevas tabelis (Tabel 3).

Tabel 3. Andmemudeli atribuutide kirjeldused.

Atribuudi nimi	**Atribuudi kirjeldus**	**Näiteväärtus**
Session_id	Sessiooni identifikaator, mille andmebaas genereerib sessiooni registreerimisel.	1234657890
Device_id	Mobiilseadme unikaalne identifikaator.	f07a13984f6d116a
Device_name	Mobiilseadme nimi.	SM-G920W8
Status	Sessiooni staatus. Võimalikud väärtused on „*active*“ (aktiivne) , „*inactive*“ (mitteaktiivne), „*expired*“ (aegunud) ja „*revoked*“ (tühistatud).	active
Activation_code	Sessiooni aktiveerimiseks genereeritud kood.	493A4-D8323-FG2A2-55BB1
Certificate_id	Sertifikaadi identifikaator, mida isik kasutas autentimiseks.	2QPH49TH65RAw4WZo4mGiYOBo4VA
Country_code	Kasutaja riiki tähistav kolmetäheline riigikood.	EST
Registration_code	Kasutaja isikukood.	47101010033
Expiration_date	Ajamärgend, mis tähistab sessiooni aegumist.	2016-04-23T18:25:43

3.3.1 Tabelite loomine

Sessiooni loomise tabeli jaoks kasutatud *SQL* lause on toodud järgneval joonisel (Joonis 16). Tabeli primaatvõti on sessiooni identifikaatorit tähistav veerg, mille väärtuse genereerib andmebaas automaatselt kirje lisamisel. Veeru väärtuse tüübiks on täisarv.

```
CREATE TABLE SESSION (
	Session_id NUMBER(10) GENERATED ALWAYS AS IDENTITY,
	Device_id VARCHAR(30),
	Device_name  VARCHAR(50),
	Status VARCHAR(30) NOT NULL,
	Activation_code VARCHAR(30) NOT NULL,
	Certificate_id VARCHAR(50),
	Country_Code CHAR(3),
	Registration_code CHAR(11),
	Expiration_date TIMESTAMP,
	CONSTRAINT Session_pk PRIMARY KEY (Session_id)
);
```

Joonis 16. Sessiooni tabeli loomine

3.3.2 Sessiooni registreerimine

Uue sessiooni loomise korral lisab veebirakendus tabelisse uue kirje, milles on määratud kasutaja andmed, aegumise ajamärgend ning staatus „*inactive*“ (Joonis 17). Andmebaas genereerib lisamisel sessiooni identifikaatori, mis on ühtlasi primaarvõti.

```
INSERT INTO SESSION (
	Status,
	Activation_code,
	Certificate_id,
	Country_code,
	Registration_code,
	Expiration_date
) VALUES (
	"inactive",
	"493A4-D8323-FG2A2-55BB1",
	"2QPH49TH65RAw4WZo4mGiYOBo4VA",
	"EST",
	"47101010033",
	"2016-04-23T18:25:43"
);
```

Joonis 17. Näide: sessiooni registreerimine.

3.3.3 Sessiooni aktiveerimine

Kui Android rakendus on teinud veebiserverile aktiveerimispäringu sobiva koodiga, aktiveerib server sessiooni. Selleks pärib esmalt server andmebaasist aktiveerimiskoodi põhjal sessiooniga seotud andmed (Joonis 18).

```
SELECT * FROM SESSION
       WHERE Activation_code="493A4-D8323-FG2A2-55BB1"
       AND Status="active";
```

Joonis 18. Näide: sessiooni andmete päring

Veebiserver uuendab vastavat sessiooni kirjet andmebaasis, lisades seadme nime ja identifikaatori ning määrates sessiooni staatuse väärtuseks „*active*“. Seejärel koostab server sessiooni andmete põhjal *JWT* ning tagastab mobiilirakendusele (Joonis 19).

```
UPDATE SESSION SET
       Device_id="f07a13984f6d116a",
       Device_name="SM-G920W8",
       Status="active",
       WHERE Activation_code="493A4-D8323-FG2A2-55BB1"
       AND Status="inactive";
```

Joonis 19. Näide: sessiooni aktiveerimine

3.3.4 Sessiooni aegumine ja tühistamine

Kui sessioon aegub, käivitub andmebaasis vastav protseduur, mis uuendab sessiooniga seonduva kirje staatust, määrates selle väärtuseks „*expired*“ (Joonis 20).

```
UPDATE SESSION SET Status="expired",
       WHERE Session_id="1234657890"
       AND Status="active";
```

Joonis 20. Näide: sessiooni aegumine.

Kasutaja välja logimisel või seadme eemaldamisel uuendab veebirakendus andmebaasis sessiooniga seonduva kirje staatust, määrates selle väärtuseks „*revoked*“ (Joonis 21).

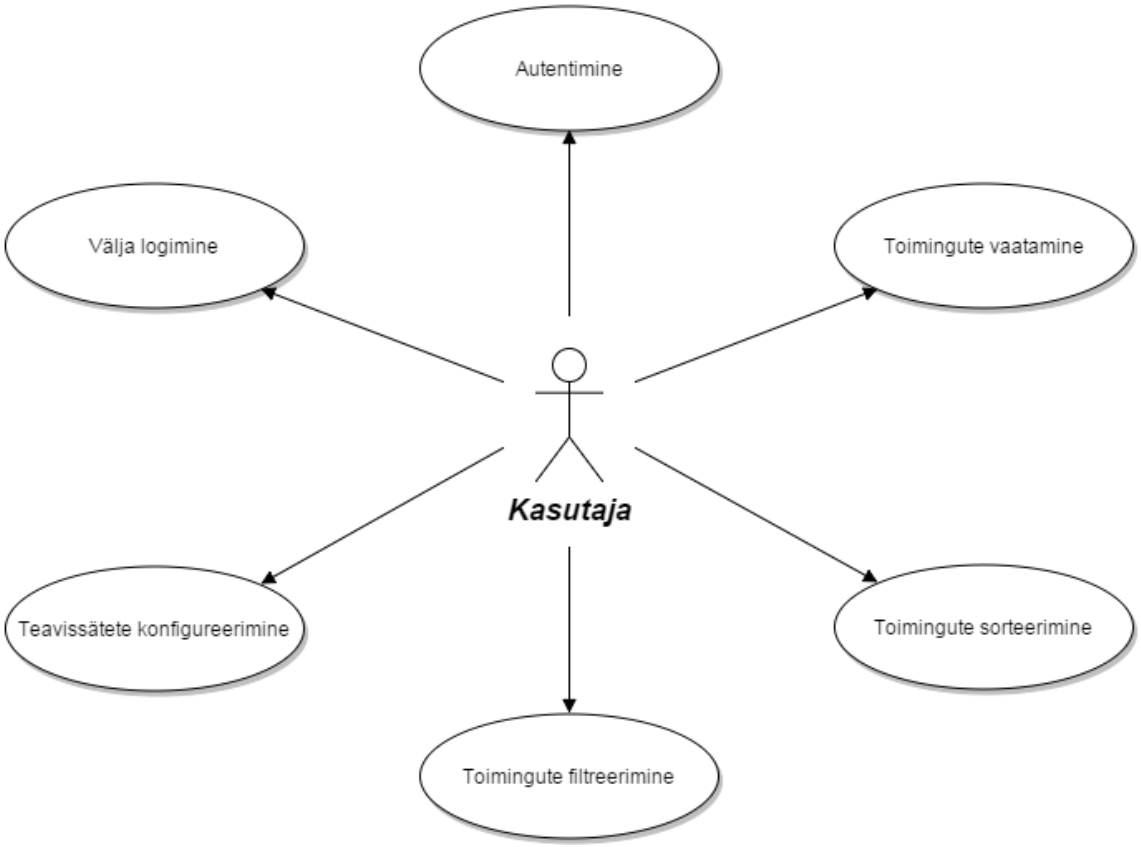
```
UPDATE SESSION SET Status="revoked",
       WHERE Session_id="1234657890"
       AND Status="active";
```

Joonis 21. Näide: sessiooni tühistamine.

4 Mobiilirakendus

4.1 Kasutusjuhud

Mobiilirakenduse kasutusjuhud on kujutatud järgneval joonisel (Joonis 22).



Joonis 22. Kasutusjuhtude mudel.

4.1.1 Autentimine

Kui kasutataval Android seadmel puudub aktiivne sessioon veebiserveriga, kuvatakse kasutajale rakendusse sisenemisel sisse logimise vaade. Sisse logimise vaates on kuvatud viide seadmete haldamise veebilehele ning tekstiväli aktiveerimiskoodi sisestamiseks. Aktiivse sessiooni korral ei ole autentimist tarvis ning kasutajale kuvatakse rakenduse avamisel „minu toimingute“ vaade.

4.1.2 Toimingute kuvamine

Kasutaja toimingud kuvatakse rakenduse peavaates nimekirjana. Nimekiri on vaikimisi sorteeritud kuupäeva järgi kahanevalt (Joonis 23).

Joonis 23. Toimingute kuvamine.

Igal toimingul on märgitud:

- **Staatus**. Kuvatud on ikoon, mis tähistab toimingul kasutatud sertifikaadi staatust. Võimalikud staatused on „*Good*“ ja „*Revoked*“. Staatusega „*Unknown*“ kasutatud sertifikaadi toimingud ei jõua rakenduseni ega vaja tähistamist.

- **Tüüp**. Sooritatud toimingu tüübid on autentimine ja digitaalne allkirjastamine. Autentimise tähistamiseks on sõrmejälje ikoon ning allkirjastamise tähistamiseks pitserdatud dokumenti meenutav ikoon.

- **Meetod**. Kuvatud on ikoon, mis tähistab toiminguks kasutatud vahendit. Toetatud vahendid on ID-kaart ja Mobiil-ID.

- **Teenusepakkuja**. Teadaoleva teenusepakkuja puhul on kuvatud vastav ikoon, vastasel juhul on kuvatud teenusepakkuja nimi tekstina.

- **Aeg**. Vorming sõltub toimingu tegemise ajavahemiku pikkusest. Varasemad toimingud on kuvatud kuupäevaga ning hiliste puhul on näidatud möödunud ajavahemiku pikkus.

4.1.3 Toimingute filtreerimine

Rakenduse sätetes on võimalik filtreerida toimingute nimekirja. Filter on rakendatav igale parameetrile. Staatuse puhul saab määrata, kas kuvada õnnestunud või ebaõnnestunud toiminguid. Tüübi korral on valik autentimiste ja allkirjastamiste vahel. Kasutatud vahendina on filtreeritavad ID-kaart ja Mobiil-ID. Teenusepakkujate filtreerimisel on valikus ainult need, mille kohta kasutajal leidub toiminguid. Saab valida ühe või mitu teenusepakkujat. Kuupäeva puhul on filtreeritav, mis ajaperioodi vahele jäävaid toiminguid kuvatakse. Määrata saab algusaja, lõpuaja või mõlemad.

4.1.4 Toimingute sorteerimine

Peavaates nimekirja kohal paiknevale toimingu parameetrile vajutades sorteeritakse nimekiri vastavalt vajutatud parameetrile (Joonis 24). Vajutus samale väljale vahetab sorteerimise järjekorda ning toimingud kuvatakse uuesti.

Joonis 24. Toimingute sorteerimine kasutatud vahendi järgi.

4.1.5 Teavitussätete konfigureerimine

Sätete vaates on seadistatav, mis tüüpi teavitusi rakendus kasutab uute toimingute avastamise korral. Teavitamine on võimalik ka välja lülitada. Kui teavitamine on aktiveeritud, ilmub vaikimisi teavitus uute toimingute korral Android seadme teavitusribal ning ikoon staatuseribal. Lisaks saab määrata, kui tihti rakendus pärib serverilt uusi andmeid kasutaja toimingute kohta. Vaikimisi on ajaintervall 10 minutit. Võimalikud valikuvariandid on: 1 minut, 5 minutit, 10 minutit, 30 minutit, 1 tund, 6 tundi, 24 tundi.

4.1.6 Välja logimine

Mobiilirakendusest välja logimine on võrdväärne veebirakenduses seadme eemaldamisega. Navigatsioonimenüüs välja logimise menüüpunkti vajutusel kuvatakse kinnitusdialoog. Kasutaja nõustumise korral saadab rakendus serverile päringu tühistada

sessioon ning kasutajale kuvatakse sisse logimise vaade. Enne uue sessiooni loomist on rakenduse kasutus piiratud sisse logimise vaate funktsionaalsustega.

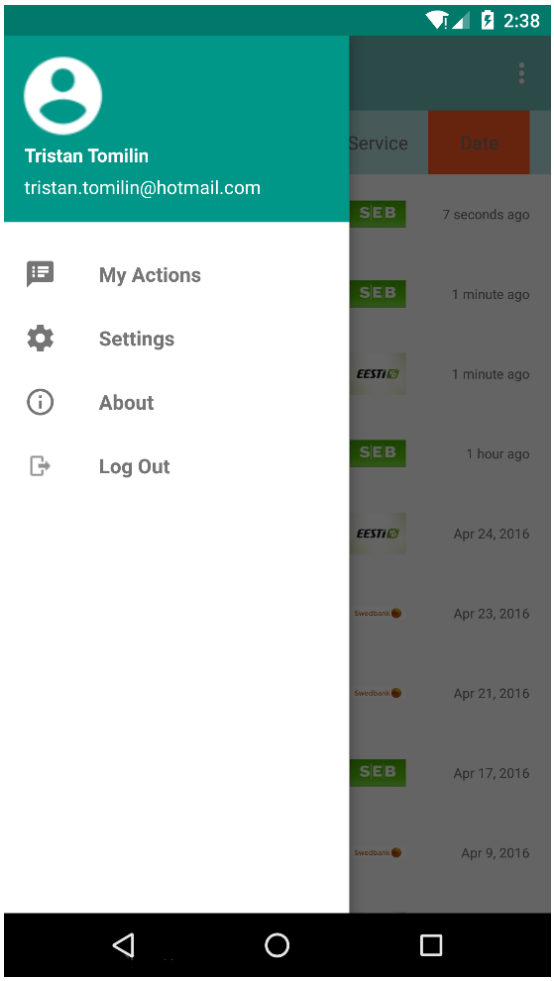
4.2 Kasutajaliides

Rakendus on disainitud *material design* [15] printsiipide põhjal. Selle metoodika kohaldamine loob rakendusele esteetilise keskkonna, mis parendab kasutajakogemust. Meeldiva visuaalse disaini korral jääb kasutajale mulje, et rakendus on usutavam ja lihtsamini kasutatav[16] . Enamikel juhtudel loob esmamulje rakenduse välimus, mitte interaktsioon [16] . Rakenduse interaktiivsed komponendid, mis lähtuvad *material design'st* on: tööriistariba, navigatsioonimenüü, diskreetsed liugurid, lülitid, kinnitusdialoogid, rippmenüüd, nupud jm.

4.2.1 Navigatsioon

Rakendusese vaadete vahel navigeerimine toimib *[15]* navigatsioonimenüü abil (Joonis 25). Navigatsioonimenüü avaneb vajutusel tööriistaribal paiknevale menüü ikoonile või vasakust ekraani servast sõrmega paremale libistades. Menüü päises on toodud aktiivse sessiooniga seotud isiku nimi ja meiliaadress. Sisu koosneb neljast menüüpunktist, mis toimivad järgmiselt:

1. *My Actions* – avab vaate „Minu toimingud".
2. *Settings* – avab vaate „Sätted".
3. *About* – avab vaate „Rakenduse info".
4. *Log Out* – logib kasutaja rakendusest välja.



Joonis 25. Rakenduse navigatsioonimenüü.

4.2.2 Toimingute vaade

Minu toimingute vaate avamisel, ilmub nimekirjete .. Toimingute nimekirja kuvamine on animeeritud. Kirjed liiguvad sorteeritud järjekorras ekraani alt vastavale kohale nimekirjas. Sorteerimise korral teostab rakendus toimingute uuesti kuvamisel sama animatsiooni. Animatsiooni kestus on pool sekundit.

Toimingute nimekirja värskendamiseks peab esmalt olemas ekraanil nähtaval nimekirja esimene element. Seejärel tuleb kasutajal libistada sõrmega ekraanil toimingute nimekirja mingist punktist alla. Selle viipe järel ilmub tööriistariba alt horisontaalselt ekraani keskele sõõrjat noolt kujutav ikoon, mis on animeeritud keerlema päripäeva. Ikoon tähistab värskendamisprotsessi. Protseduuri lõppedes kahaneb ikoon oma keskpunkti. Uute toimingute olemasolu korral lisab rakendus need nimekirja algusesse, rakendamata animatsiooni, mis on kasutusel kogu nimekirja kuvamisel.

4.2.3 Rakenduse sätted

Rakenduse sätetesse pääseb navigatsioonimenüüst vajutades menüüpunktil *Settings*. Sätted on teavituste konfigureerimiseks ja toimingutele filtrite rakendamiseks.

Teavitusmeetodite aktiveerimiseks on kasutusel lülitid. Teavitusintervalli määramine toimub rippmenüü abil.

Toimingu staatuse, tüübi ja kasutatud vahendi filtri aktiveerimine on realiseeritud raadionuppude abil. Nimetatud toiminguparameetritel on kaks võimalikku väärtust, mis on kuvatud ka raadionupu väärtuse valikuna vastavalt rakendatavale filtrile. Vaikimisi on aktiivne valik väärtusega „Väljas“. Filtri kohaldamiseks on tarvis valida erineva väärtusega raadionupp.

Teenusepakkuja filter aktiveerub lülitiga. Vajutuse järel kuvatakse kasutajale hüpikaknana nimekiri teenusepakkujatest, kelle kohta isikul toiminguid leidub. Filter on rakendatav nii ühele kui ka mitmele teenusepakkujale. Valiku tühistamise korral jääb filter mitteaktiivseks.

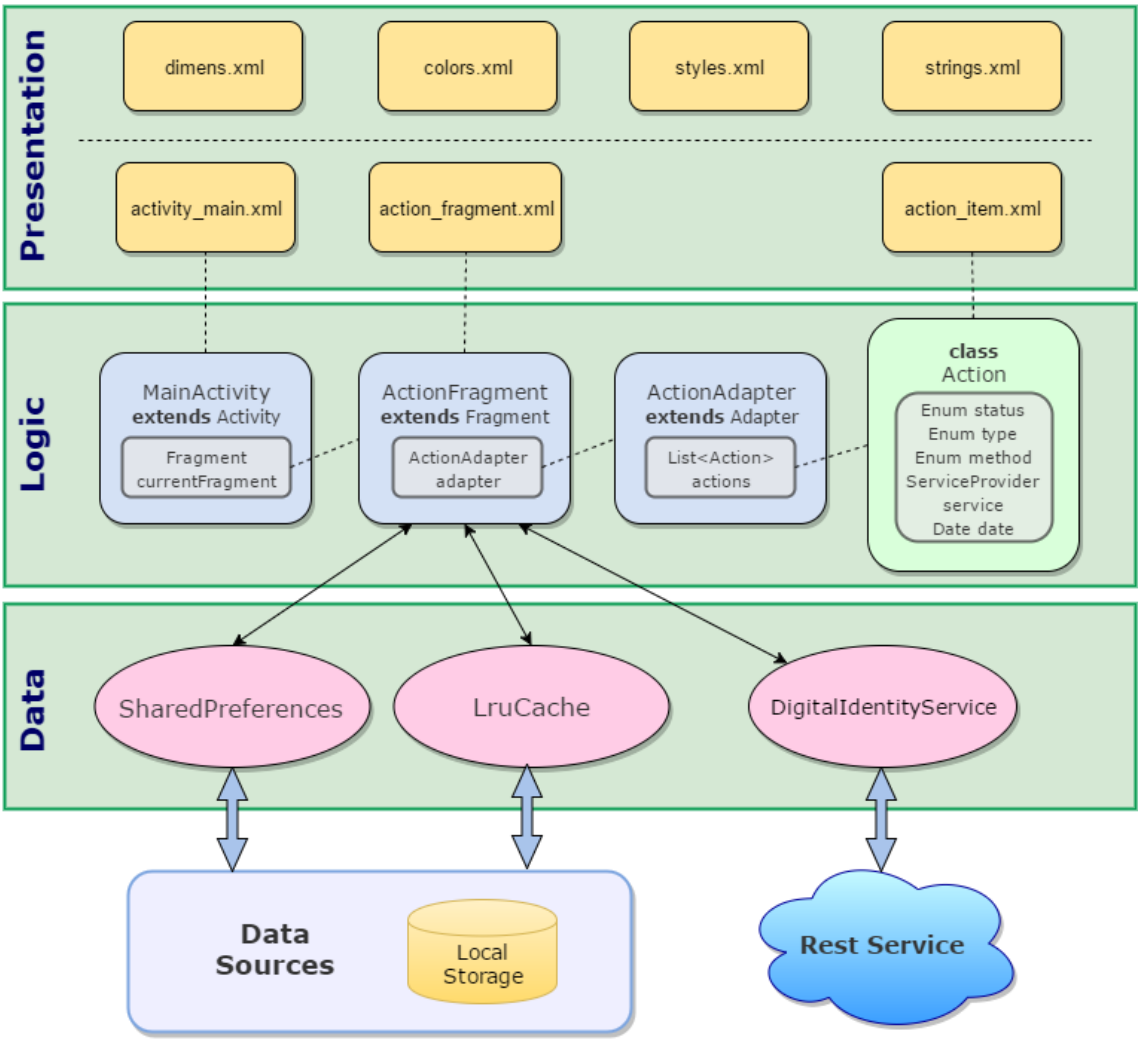
Aja filter on rakendatav vastavast lülitist. Lüliti aktiveerimisel ilmub kuupäevavalija. Kuupäeva määramise järel ilmub kellaajavalija. Kellaaja määramisel aktiveerub filter. Kumbagi valiku tühistamisel muutub lüliti taas mitteaktiivseks ning toimingute filtreerimist ei rakendata.

4.2.4 Võrguühendus

Rakenduses toimingute nimekirja värskendamiseks ja teavituste saamiseks on tarvis, et mobiilseadmel oleks andmeside lubatud ning toimiv võrguühendus. Rakenduse kasutamine *offline* režiimis on piiratud. Avamise korral kontrollib rakendus sessiooni kehtivust tehes serverile vastava päringu (Joonis 10). Ühenduse puudumise korral sisaldab *HTTP* päringu vastus veakoodi 503 (*Service Unavailable*) [14] ning rakenduses kuvatakse kasutajale teade võrguühenduse puudumisest. Kui kasutaja üritab värskendada toimingute nimekirja, saadab rakendus serverile kasutaja viimaste toimingute päringu (Joonis 12), mis võrguühenduse puudumisel saab vastuseks samuti veakoodi 503 ning kasutajale kuvatakse sama teade. Ülejäänud funktsionaalsus ei sõltu ühenduse olemasolust. Rakenduses saab kuvada olemasolevaid toiminguid, neid sortreerida ja filtreerida ning konfigureerida sätteid.

4.3 Arhitektuur

Rakenduse arhitektuuri mudel koosneb kolmest kihist (Joonis 26).



Joonis 26. Rakenduse kihiline arhitektuur.

Esitlus- ehk presentatsioonikihi moodustavad *xml* formaadis failid, mis jagunevad otstarbe järgi kaheks. Ühte tüüpi failid sisaldavad rakenduse ainestikule vastavaid elementide väärtuseid ja teised küljendavad rakenduse vaateid, määrates kuvatavate elementide asukoha, visuaalse kuju ja käitumise. Esitluskiht registreerib kasutaja tegevust ning edastab informatsiooni loogikakihile [17] . Saabuvad andmed loogikakihilt kujundatakse kasutajale nähtavateks elementideks. Esitluskiht moodustab kasutajaliidese.

Loogikakiht käsitleb esitluskihilt saadud päringuid ning vajadusel tagastab väljundi [17] . Loogikakihi ühe osa moodustavad andmemudelid ja olemitüübid, mis kapseldavad vajalikud andmed ja äriloogika, et esindada pärismaailma elemente nagu isiku digitaalsed toimingud [18] . Defineeritud andmeatribuutide tüübid tagavad, et hoitud andmed oleks äriloogikale sobival kujul. Mudelite abil opereerivad teist tüüpi loogikakihi elemendid, mis korraldavad protseduurilist tööd. Need komponendid määratlevad kindla järjestusega äriloogika protseduurid, mis rakendatakse teistelt kihtidelt vastu võetud päringute korral. Loogikakiht juhib rakenduse tööd, vastutades andmete sobivuse ja kehtivuse eest.

Andmekihi moodustavad komponendid, mille üks otstarve on pärida andmeid seadme sisemälust ja välistelt teenustelt ning tagastada need loogikakihile. Teiseks otstarbeks on salvestada loogikakihilt saadud andmed seadme sisemällu või edastada välistele teenustele. Andmekiht kontrollib vahendatavate andmete vastavust ettenähtud formaadile. Primitiivsete andmete talletamiseks lokaalses mäluhoidlas on kasutusel *SharedPreferences*. Selliste andmete hulka kuulub *JWT*, mis on vaja toimingute pärimisel saata serverile, et sessioon identifitseerida. Teenuse *DigitalIdentityService*'i abil päritakse rakenduse väliseid andmeid kasutaja toimingutest, kasutades *Retrofit* raamistiku võimalusi. Toimingute vahemällu salvestamisega ja sealt lugemisega tegeleb utiliit *LruCache*.

Rakenduse eri vaadete loomiseks on kasutatud fragmente. Fragmentidel on defineeritud eraldi elutsükli etapid, mis võimaldab laiemat valikut, mis hetkel protseduure rakendada. Fragmendid on taaskasutatavad, mis muudab toimingute vaate korduva laadimise hõlpsamaks rohkete kirjete korral. Lisaks on paremini realiseeritud rakenduses tagasi liikumine. Tavaliste vaadete korral võib näiteks orientatsiooni korduval muutmisel kasutaja tegevuste ajalugu rikneda.

5 Kokkuvõte

Bakalaureusetöö eesmärgiks oli isiku ID-kaardi ja Mobiil-ID digitaalsete toimingute teavitamisrakendusest ülevaate loomine. Töö raames kirjeldatud mobiilirakendus pakub osalist lahendust digiteenuste turvalisuse probleemile, teavitades kasutajat tema identiteediga seotud toimigute korral.

Projekti raames kasutatud tehnoloogiad tõstavad kogu süsteemi turvalisust ning parendavad rakenduse kasutust. Mobiilseadmes töötava rakenduse ja serveri vahelise sessiooni loomiseks on kolmanda osapoolena kasutusel veebirakendus, mis nõuab kasutajalt turvalise vahendiga autentimist. Kuna mobiilseade ei ole piisavalt turvaline, toimub ka seadmete haldus mobiilirakendusest sõltumata.

Töö tulemusena valmis ülevaade digitaalsete toimingutega kaasnevatest turvaohtudest ning rakenduse otstarbest nende tagajärgede leevendamiseks. Loodud on kirjeldus teenustest, mille kaudu toimub mobiilirakenduse ja serveri andmevahetus ning päringute autentsuse kontrollimiseks rakendatud protseduure. Ühtlasi on antud ülevaade andmemudelist ja kirjeldatud käsud, mida veebiserver kasutab andmebaasist päringute tegemiseks ja sinna andmete talletamiseks. Meeldiva kasutajakogemuse pakkumiseks, on mobiilirakenduse kasutajaliidese loomisel rakendatud kindlaid disainimisprintsiipe ja -mustreid. Loodud on ülevaade mobiilirakenduse kasutusjuhtudest ja võimalikest funktsionaalsustest. Kirjeldatud on Android rakenduse arhitektuuri kihte ning nende omavahelist sõltuvust.

Projekti edasi arendamise võimalus oleks siduda server X-tee kaudu rohkemate teenustega. Selle korral mobiilirakenduse teavitamisvaldkond laieneks. Kasutajale laekuksid teavitused ka näiteks tema isikuandmete muutmise ja käitlemise korral või temaga seotud dokumentide loomise, muutmise, aegumise ja tühistamise korral. Kui isik on pidevalt kursis temaga seotud digitaalse maailma protseduuridest, vähenevad ka turvariskid.

Kasutatud kirjandus

[1] Certificate Centre Ltd. [Online]. Available: https://sk.ee/ [Accessed 6 May 2016].

[2] Wikipedia, "Representational State Transfer", [Online]. Available: https://en.wikipedia.org/wiki/Representational_state_transfer [Accessed 6 May 2016].

[3] Wikipedia, "ID-kaart", [Online]. Available: https://et.wikipedia.org/wiki/Eesti_ID-kaart [Accessed 6 May 2016].

[4] "E-teenused, kus saad ID-kaarti ja Mobiil-ID-d kasutada", [Online]. Available: http://www.id.ee/index.php?id=30230 [Accessed 6 May 2016].

[5] Jacob A., "Küberkuritegevus", [Online]. Available: http://www.arvutikaitse.ee/arvutikaitse-algtoed/kuberkuritegevus/ [Accessed 8 May 2016].

[6] Jacob A., "Privaatsus", [Online]. Available: http://www.arvutikaitse.ee/arvutikaitse-algtoed/privaatsus/ [Accessed 8 May 2016].

[7] Jacob A., "Trikitamine", [Online]. Available: http://www.arvutikaitse.ee/arvutikaitse-algtoed/trikitamine/ [Accessed 8 May 2016].

[8] Riigikogu, "Isikuandmete kaitse seadus", [Online]. Available: https://www.riigiteataja.ee/akt/130122010011 [Accessed 8 May 2016].

[9] Riigi Infosüsteemi Amet, "X-tee tutvustus", [Online]. Available: https://www.ria.ee/ee/x-tee-tutvustus.html [Accessed 8 May 2016].

[10] Square, Inc., "Retrofit Introduction", 2013, [Online]. Available: http://square.github.io/retrofit/ [Accessed 12 May 2016].

[11] Auth0, "Introduction to JSON Web Tokens", [Online]. Available: https://jwt.io/introduction/ [Accessed 12 May 2016].

[12] Defuse Security, "Salted Password Hashing – Doing it Right", April 13, 2016, [Online]. Available: https://crackstation.net/hashing-security.htm [Accessed 13 May 2016].

[13] "SharedPreferences", [Online]. Available: https://developer.android.com/reference/android/content/SharedPreferences.html [Accessed 13 May 2016].

[14] RestApiTutorial.com, "HTTP Status Codes", [Online]. Available: http://www.restapitutorial.com/httpstatuscodes.html [Accessed 15 May 2016].

[15] Google, "Navigation drawer", [Online]. Available: https://www.google.com/design/spec/patterns/navigation-drawer.html [Accessed 17 May 2016].

[16] Gócza Z., "Myth #25: Aesthetics are not important if you have good usability", [Online]. Available: http://uxmyths.com/post/1161244116/myth-25-aesthetics-are-not-important-if-you-have-good-us [Accessed 17 May 2016].

[17] Savolainen M., "Arhitektuur", [Online]. Available: http://ats.cs.ut.ee/tvp2009/php/st/dist/st-iter2-snapshot/doc/modelling/architecture.pdf [Accessed 17 May 2016].

[18] Micrsoft, "Chapter 7: Business Layer Guidelines", 2016, [Online]. Available: https://msdn.microsoft.com/en-us/library/ee658103.aspx [Accessed 19 May 2016].

[19] Micrsoft, “Chapter 8: Data Layer Guidelines”, 2016, [Online]. Available: https://msdn.microsoft.com/en-us/library/ee658127.aspx [Accessed 19 May 2016].

[20] Wikipedia, “QR-kood”, March 9, 2015, [Online]. Available: https://et.wikipedia.org/wiki/QR-kood [Accessed 21 May 2016].

[21] Cybernetica, “X-tee”, [Online]. Available: http://cyber.ee/e-riik/x-tee/ [Accessed 21 May 2016].