Tallinna Pedagoogikaülikool
Matemaatika-loodusteaduskond
Informaatika osakond

Mati Vapper

Eesti Äriregistri sidumine ID-kaardiga

Diplomitöö

Juhendaja: Jaagup Kippar

Autor: ..... “……”.....2002
Juhendaja: ..... “…….”.....2002
Osakonna juhataja:..... “…….”.....2002

Tallinn 2004

Sisukord

Sissejuhatus

Eesti Äriregister on arvatavasti üks suurimaid infosüsteeme Eestis. Seda on arendatud juba alates 1996. aastast. Mõne aasta eest hakkas äriregister tööle ka internetis, kust sai teha lihtpäringuid ja detailpäringuid Eestis tegutsevate äriühingute kohta. Umbes samal ajal sai äriregister ligipääsetavaks ka läbi WAP keskkonna e. mobiiltelefonidele. Nüüd, kus informatsiooni õigeaegne kättesaamine on muutunud maailmas ülimalt oluliseks, on Eesti Äriregister laienenud ka X-Tee kodanikuportaalini. Õige pea on valmimas üleeuroopaline projekt “Euroopa Äriregister” e. EBR, kust saab samuti teha päringuid Eesti asutuste kohta. Eesti Äriregistri veebikeskkond on üsna mitmekülgne ja võimaldab teha mitmesuguseid operatsioone – alates lihtpäringutest ja lõpetades ettevõtete majandusaasta aruannete sisestamisega. Lihtpäringud on kõigile tasuta, need sisaldavad endas ettevõtete üldandmeid (nimetus, aadress, aktsia-/osakapital). Eesti Vabariigi seadusandlus lubab kõikidel kodanikel teha ka detailpäringud asutuste kohta, kuid need on tasustatud.

Eesti on digitaliseerimisel väga eesrindlik riik. Siinkohal pean silmas siinjuhul eelkõige seadusandlusega kinnitatud digitaalallkirjastamist ja isikutunnistusi, kus digiallkirjal on samad õiguslikud tagajärjed kui käsitsi kirjutatud allkirjal ja elektrooniline isikutunnistus e. ID-kaart on muudetud kohustuslikuks Eesti Vabariigi kodanikele. Äriregister kuulub justiitsministeeriumi registrikeskusele, kes vastutavad terve süsteemi toimimise eest. Äriregistri parandusi ja arendustöid tellitakse Realsüsteemid AS-st pea igal aastal. Äriregistri infosüsteem, nagu ka paljud teised riiklikud süsteemid, vajab pidavalt arendamist, kuna seadustik muutub ja seadusi tuleb juurde.

Töö struktuur on etappide kaupa jaotatud, teemad koosnevad enamasti sissejuhatavast osast, töö iseloomust, kirjeldusest, Äriregistri probleemidest/lahendustest, seletavatest skeemidest ja näiteprogrammidest.

Antud projekt on praktiline töö ja hõlmab endas eeskätt äriregistri veebikeskkonna arendustöid. Veebikeskkond on tihedalt seotud ka äriregistri keskandmebaasiga, mis on omakorda seotud kõikide osakondade andmebaasidega. Töö põhieesmärgiks on luua turvaline veebikeskkond, mis võimaldaks ettevõtetel sisestada registrisse ja digiallkirjastada majandusaasta aruandeid, uuendada või sisestada ettevõtte sidevahendite loetelu ja kuna äriregistris on juriidiliste isikutena registreeritud ka erakonnad, siis oli tarvis võimaldada erakondadel liikmete nimekirju veebist sisestada ning neid digitaalselt allkirjastada. Sarnane toimiv süsteem oli ennegi olemas, ainult, et ilma digiallkirjastamise ja ID-kaardiga autentimise võimaluseta. Lisaks sellele peab uus veebikeskkond suutma teostada autentimist ID-kaardiga, kasutades OCSP (Online Certificate Status Protocol) protokolli; võimaldama sisestatud DDOC (digitaalselt allkirjastatud dokumendi või dokumentide komplekti formaat, mida kasutab DigiDoci signeerimisprogramm) failides olevate digiallkirjade valideerimist; analüüsima DDOC faili sisu ja kirjutama selle andmebaasi; digiallkirjastama allkirjastamata majandusaasta aruannete faile ja toimetama allkirjastatud failid allkirjastajale koos asutuse sidevahendite ja osanike/aktsionäride nimekirjaga, mis allkirjastatakse koos majandusaasta aruande failiga.

Digiallkirjastamine äriregistri veebikeskonnas lihtsustab oluliselt registri osakondade tööd. Andmete kogus on küllaltki suur ja seepärast on ka nende töötlemine mahukas töö.

Teist taolist süsteemi Eestis tehtud ei ole, kuid sarnaste võimalustega lahendus digidoc'i failitöötluse vallas on Sertifitseerimiskeskuse DigiDoci portaal (http://digidoc.sk.ee). ID-kaardiga autentimist võime näha juba paljudes kohtades (pankade klienditeeninduskeskkondades, portaalides, mobiili operaatorite virtuaalsetes teenindusbüroodes jne.). Enamasti kasutatakse autentimiseks tühistusnimekirju, mitte OCSP-d. Tühistusnimekirjade puuduseks on see, et neid uuendatakse teatud aja tagant ning seega ei ole võimalik teha reaalajas autentimist, ehk autentimine kehtetud kaardiga võib saada teatud ajahetkel võimalikuks.

Käesolevas projektis on kasutatud digiallkirjastamiseks, autentimiseks ja teisteks taolisteks operatsioonideks sertifitseerimiskeskuse tellimusel valmisprogrammeeritud Java DigiDoc-i teeke, mida käivitatakse läbi veebiserveris asetsevate java

programmidega. Java arendustööriistaks on kasutatud JDK 1.3.3 tarkvara. Äriregister on juba algusest peale asetsenud Oracle andmebaasides. Ka äriregistri veeb asetseb terviklikult andmebaasis – brauseri jaoks käivitatakse baasiprotseduure, mis lõppkasutajale veebi sisu näitavad. Oracle baasiprotseduurid on kõik PL-SQL keeles, mida antud töös kirjutasin Oracle SQL+ tarkvaraga. Töös o Oracle SQL+ tarkvaraga. Töös oon modifitseeritud ja lühendatud ülevaatlikkuse huvides.

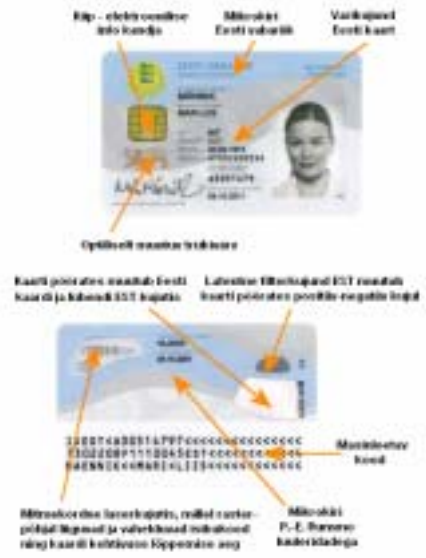
1 ID-kaart ja digiallkiri

Ka mujal maailmas on läbi viidud väga palju PKI-pilootprojekte (Public Key Infrastructure) nii era- kui ka avalikus sektoris ning kulutatud sellele hulk raha. Tänaseks on jõutud seisu, kus erinevates ettevõtetes töötavad oma firmakesksed infrastruktuurid ning sertifikaatide ja neid kandvate kaartide rakendused piirduvadki firmasiseste lahendustega. Nende infrastruktuuride ühendamisega ollakse suures hädas ja alles otsitakse vastuseid.

Eesti on valinud muust maailmast erineva tee, luues üleriikliku infrastruktuuri ja kaardi, mis on mõeldud kasutamiseks kõigis uutes ja olemasolevates rakendustes nii era- kui ka avalikus sektoris. Samuti on Eesti digitaalallkirja seadus võrreldes muu maailma vastavate seadustega on suhteliselt väikese mahuline ja hästi rakendatav, kuid vastab sisu poolest täielikult Euroopa Liidu seadusandlusest tulenevatele nõuetele.[1] (joonis 1)

Eesti ID-kaart on kiipkaart, mis sisaldab endas kahte sertifikaati: autentimissertifikaat ja signeerimissertifikaat. Mõlemad sertifikaadid sisaldavad järgmisi andmeid:

- sertifikaadi väljaandja andmed (nimi, registrikood)
- sertifikaadiomaniku andmed (eesnimi või -nimed, perekonnanimi, isikukood);
- sertifikaadi kehtivusandmed (moodustamise kuupäev ja kellaaeg, aegumise kuupäev ja kellaaeg)
- muud sertifikaadipõhised andmed



joonis 1. ID-kaart

[1] www.id.ee

1.1 Eeltöö ja taustauuring

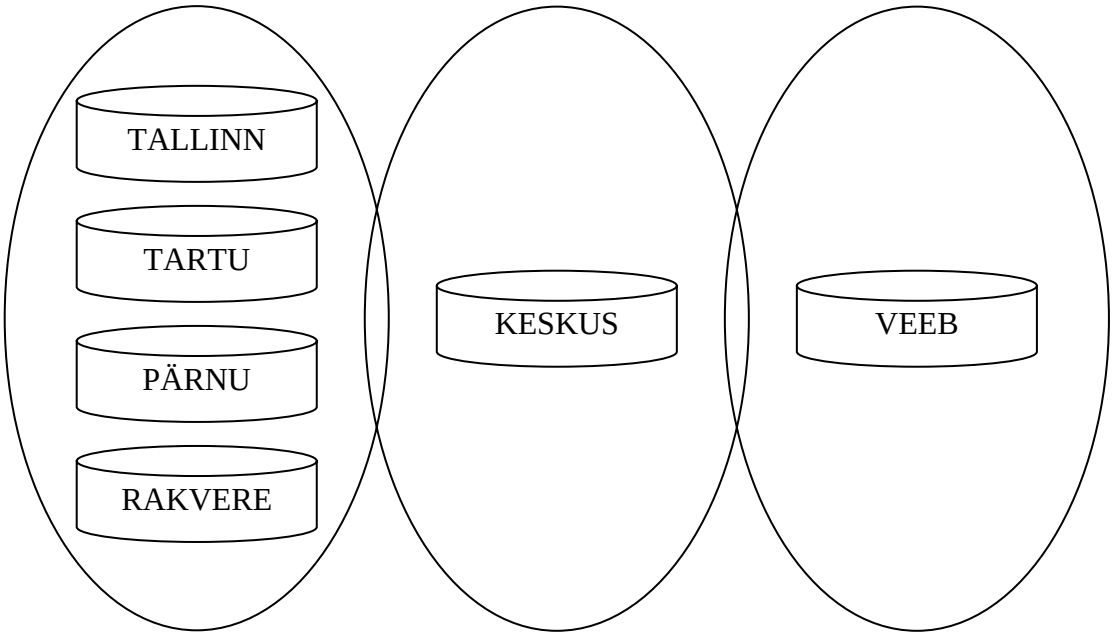
Esmalt oli antud projekti juures vaja ID-kaarti. Isiklikku ID-kaarti puudumise tõttu ei tuli Sertifitseerimiskeskusest taotleda test ID-kaart. Test ID-kaart on Mari-Liis Männiku nimele registreeritud kaart, mis pole seotud ühegi reaalse isikuga. Kahjuks polnud Mari-Liis Männiku ID-kaardil kehtivaid sertifikaate, millega testida OCSP toimivust kehtiva kaardina. Lõppkokkuvõttes polnud sellest erilist probleemi, kuna kehtivate sertifikaatidega ID-kaarte oligi tunduvalt lihtsam leida kui kehtetuid. Vähemalt õnnestus testida järgmisi situatsioone: mis juhtub kehtetu kaardiga autentimisel ning kas süsteem toetab täpitähti. Täpitähtede tugi kummalisel moel esialgu ei töötanudki.

Olles pidevas kontaktis Eesti ID-kaardi juurutajatega ja arendajatega, jõudis autor järelduseni, milliseid vahendeid antud situatsioonis on vajalik kasutada. Eestis on allkirjastamiseks ja muudeks ID-kaardiga seotud toiminguteks loodud programm nimega DigiDoc. Digidoc on lihtne kohalik programm, mis kasutab C-s kirjutatud teeke. Sisuliselt sarnane programm nagu DigiDoc tuleb igal veebiprogrammeerijal, kes soovib oma saidile ID-kaardi tuge luua koos allkirjastamisega, ise valmis programmeerida, kasutades kas neid samu C-s kirjutatud teeke või veel lisaks loodud Java DigiDoci teeke. Üle interneti autentimiseks ja allkirjastamiseks on seega vaja kasutada veebiserveripoolseid programme, mis kasutavad SEB-grupi poolt valmistatud Java ja/või C teeke, mis omakorda kasutavad mitmeid kokku komplekteeritud turvateeke, xml-parsereid ja palju muud. Eestis on erinevates portaalides kasutatud mõlemaid variante. Näites Ühispanga kasutajaportaalis on kasutatud Java teeke, aga Sertifitseerimiskeskuse DigiDoc'i portaalis on kasutatud C teeke. Autor jõudis arusaamani, et Äriregistri situatsiooni juures oleks kõige mugavam kasutada Java DigiDoc'i teeke. Põhjused olid järgmised: Eesti Äriregistri veebiserveris on töötav Jakarta Tomcat (Java Application Server); Java kood on autorile vastuvõetavam, kui C kood; Äriregistri veebiserveris on tehtud teisigi Java lahendusi (näiteks isiku autentimine läbi pankade);

2 Äriregistri infosüsteemi ülesehitus

2.1 Oracle andmebaas

Eesti Äriregister töötab tervikuna Oracle andmebaasidel. Baasid jagunevad kolmeks: osakondade baasid, keskbaas ja veebibaas (joonis 2.). Äriregistri osakondi on hetkel 4:



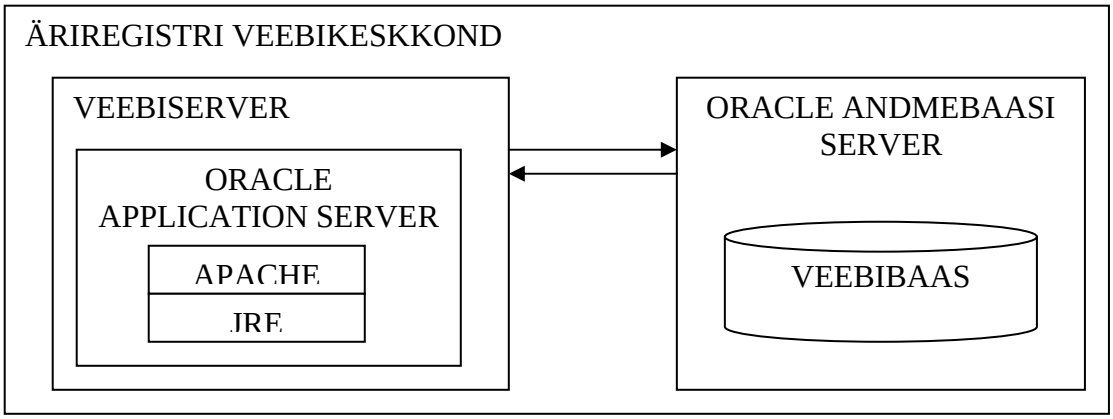
joonis 2. andmebaaside plaan

Tallinna registriosakond, Tartu registriosakond, Pärnu registriosakond ja Rakvere registriosakond. Kõikides osakondades on oma andmebaasid ja sisuliselt on andmed kõik dubleeritud ka keskregistri osakonna andmebaasis. Taoline struktuur hoiab ise enda jaoks tagavara koopiat ning samas on suuteline töötama sõltumatult teistest baasidest. Näiteks, kui midagi peaks juhtuma keskbaasiga, saavad osakonnad jätkata endist tööd senikaua kuni keskbaas jälle tööle hakkab. Andmebaasid on üksteisega väga tihedalt seotud. Näiteks, kui peaks toimuma muudatus veebibaasis, siis on oluline, et need andmed jõuaksid kiiresti keskusesse ja sealt vastavatesse osakondadesse ja vastupidi. Näide võiks olla järgmine: isik logib end Äriregistri veebikeskkonda ja muudab oma asutuse sidevahendite loetelu, misjärel kohe või mõne aja möödudes käivitatakse keskbaasis protseduur, mis sünkroniseerib sidevahendite tabeli veebi-

,keskbaasi ja osakondade andmebaaside sisu. Laeb uued andmed keskele ja saadab vastavatesse osakondadesse, kus andmeid edasi juba käsitsi töödeldakse, misjärel käivitatakse jälle protseduurid, mis osakondades tehtud muudatusi kesk- ja veebibaasis kajastavad.

2.2 Iseseisev veebiserver

Veebiserver on tihedalt seotud veebibaasiga. Oracle keskkonnas on laialt levinud Oracle Application Serveri tarkvara, mis sisaldab muuhulgas endas Oracle Jdeveloperi – Java arendustarkvara, JRE –d (Java Runtime Environment), JDK-d (Java Developer Kit), veebiserverina on Oracle Application Serveri komplektis Apache'i serveri tarkvara jms. (joonis 3.)

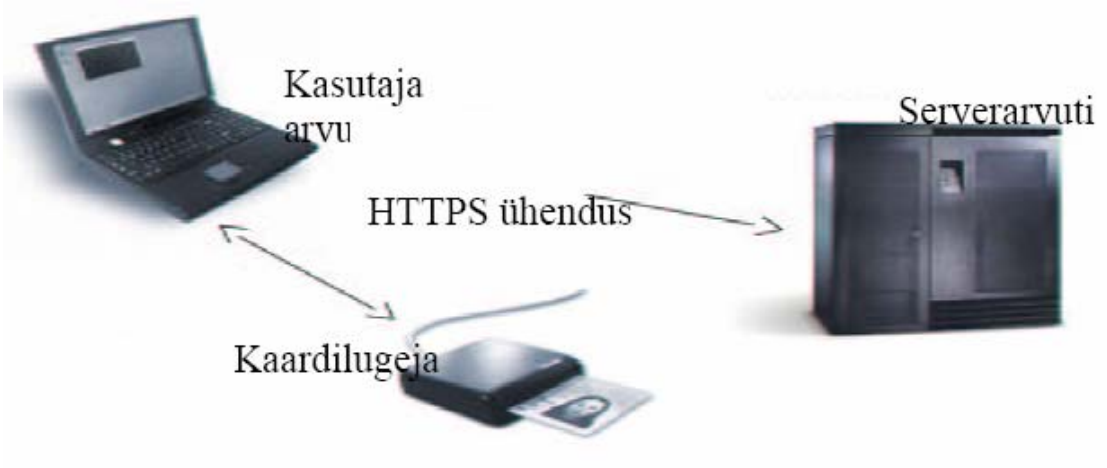


joonis 3. serverite vahelised suhted

Äriregistri veebikeskond on dünaamiline süsteem, kus pea kõik komponendid ja andmed asuvad andmebaasis. Visuaalse poole pealt on ainsana staatiline Äriregistri veebikeskkonna pealeht, millelt suundutakse edasisi toiminguid teostama. Kõik muu, mis edaspidi kasutajale näha on asetseb andmebaasis. Julgen isegi arvata, et see on üks väheseid veebi liideseid, kus ka protseduurid, mis veebi kuvavad on kõik andmebaasis. Vaid väike osa protseduuridest, mida tehnilistel põhjustel pole olnud võimalik Oracle baasi integreerida, asetsevad eraldi veebiserveris, mida Oracle protseduuridest välja kutsutakse. Ka ID-kaardi tehnoloogiat puudutavad programmid, on enamasti eraldi Oracle andmebaasist. Ainult, et neid ei kutsuta otseselt välja Oracle baasist, aga sellest juba natuke hiljem.

2.3 Seadistus

Serveri seadistamisel tuleb lähtuda kasutusel olevatest võimalustest ja turvadirektiividest. Äriregistri projekti juures on veebiserver seadistatud töötama järgmiselt: (joonis 4.)



joonis 4. serveri seadistus

JdigiDoci seadistamine tugineb suures osas Veiko Sinivee koostatud juhendile:
JDigiDoc on programeerimiskeeles Java loodud teek. Antud teek pakub funktsionaalsust DIGIDOC-XML 1.1 formaadis digitaalselt allkirjastatud failide loomiseks, lugemiseks, allkirjastamiseks, kehtivuskinnituse hankimiseks ja allkirjade ning kehtivuskinnituste kontrolliks. JDigiDoc klassid järgivad küllaltki täpselt XML-DSIG ja ETSI standardit ja pakuvad mugavat kasutajaliidest antud objektide loomiseks ja kasutamiseks.[2]

JdigiDoci teek on sõltuv veel mitmetest teistest teekitest, mis tuleb ka kõik serverisse installeerida enne kui JdigiDoc'i kasutama saab hakata.

JDigiDoc teek sõltub järgmistest komponentidest:

[2] Veiko Sinivee, JdigiDoc'i kasutusjuhend

- Java2 – JDK/JRE 1.3.1 või uuem
- Apache XML Security – Vajalik kanoniseerimiseks
- XML parser – Apache Xerces. Vajalik Apache XML Security teegi jaoks.
- Xalan – Versioon 2.2D13 või uuem. Vajalik Apache XML Security teegi jaoks.
- IAIK JCE krüptoteek – Vajalik IAIKNotaryFactory klassis kehtivuskinnituste koostamiseks ja parsimiseks. Kui see factory välja vahetada siis pole IAIK JCE teeki vaja.
- Bouncy-Castle krüptoteek – Vajalik PKCS11DigiDocFactory klassis krüptograafilisteks operatsioonideks. Sobiks tegelikult suvaline Java krüptoteek. Valiti Bouncy-Castle teek, kuna see on vabavara teek.
- Jakarte Log4j - Vajalik Apache XML Security teegi jaoks.[3]

[3] Veiko Sinivee, JdigiDoc'i kasutusjuhend

3 Autentimine

ID-kaardi üks peamisi funktsioone on isiku kindlaks määramine. See on Eestis ainus isikut tõendav dokument, millega on digitaalselt võimalik kindlaks teha isik, kes antud kaarti kasutab (näiteks panka sisse logimine, X-tee päringute teostamine enda kohta, virtuaalne arvete maksmine jne.). Selleks, et autentimist teostada, peab konkreetne server võtma ühendust teise serveriga, kus on kirjas autenditava isiku autentimissertifikaat ning selle sertifikaadi aegumiskuupäev. Turvalise autentimise huvides on välja töötatud hulk lahendusi. Siinkohal on tutvustatud neist kahte.

3.1 Autentimine tühistusnimekirjade (CRL) alusel

Tühistusnimekirjad e. CRL (Certificate Revocation List) failid on tehnoloogia mis võeti kasutusele umbes samal ajal, kui hakati kasutama PKI (Public Key Infrastructure) süsteemi, ehk avaliku võtme infrastruktuuri. Avaliku võtme krüptoalgoritm leiutati 1970-ndate aastate lõpus ning seda nimetatakse ka asümmeetriliseks krüptograafiaks. See süsteem lubab kasutajatel vahetada turvaliselt andmeid mitteturvalises võrgus (näit. internetis), kasutades krüptograafiliselt loodud võtmepaari kasutaja avalikust võtmest ja privaatvõtmest. Sealjuures võtmepaar luuakse seadusega määratud organite poolt. PKI sisaldas endas ka kasutajate sertifikaate, millega saab identifitseerida indiviidi või asutust. Sertifikaatide puhul on olulisel kohal nende kehtivus. Organ, kes sertifikaate väljastab, määrab ära nende aegumiskuupäeva. Samas võib see võimuorgan muuta erandkorras sertifikaadi kehtetuks, näiteks ID-kaardi kaotamisel, vältida selle väärkasutamist.

CRL-il baseeruv süsteem peab võtma ühendust serveriga, kus asuvad sertifikaatide kehtivuse olekud, ehk server, mis teostab autentimist, peab laadima alla iga teatud aja tagant kehtiva sertifikaatide seisu. See süsteem pole väga otstarbekas, antud olukorras, kuna võib tekkida ajahetk, kus autentimisserveris on aegunud andmed ja see võib

tekitada süsteemi suure turvaaugu. Parem autentimissüsteem on autentimine üle OCSP teenuse.

3.2 OCSP kehtivuskinnituse teenus

OCSP (Online Certificate Status Protocol) kehtivuskinnituse teenus on avalik teenus, mida Eestis (nagu ka muid ID-kaardiga seotud teenuseid) pakub AS Sertifitseerimiskeskus. OCSP protokoll on samm edasi CRL-st, kuna kaob ära probleem, kui serveris, mis tegeleb sertifikaatide autentimisega, on aegunud situatsioon. Samuti pole OCSP puhul tarvis pidevalt alla laadida CRL-faile. OCSP kontrollib reaalajas kliendi sertifikaatide olekut ja saadab veebiserverile vastuse.
Vastusetüübid:

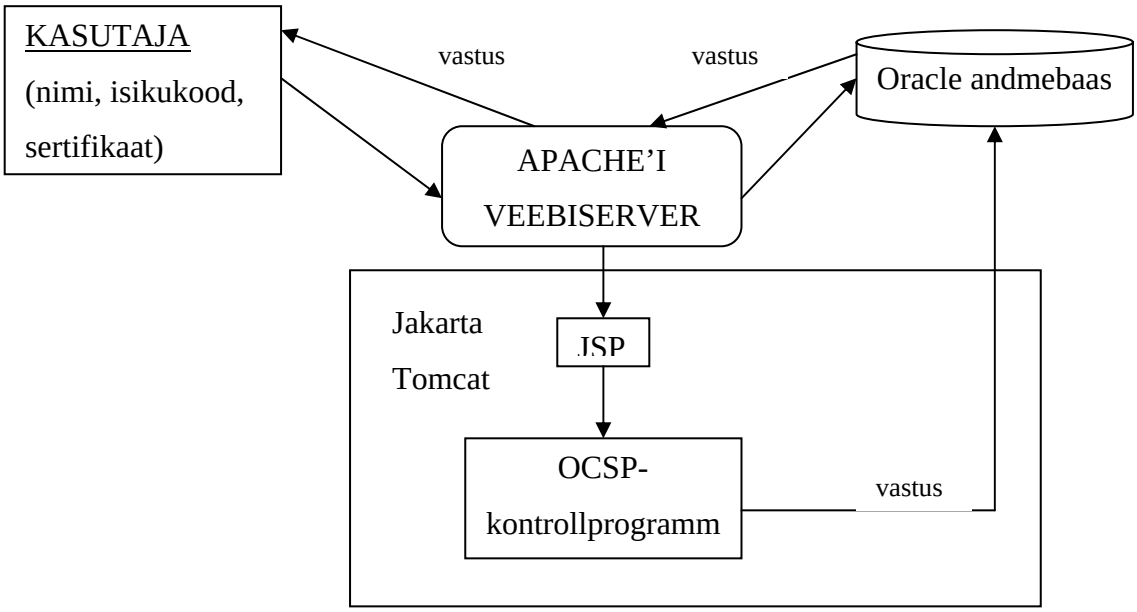
- Kehtiv olek (current)
- Kehtetu olek (expired)
- Tundmatu olek (unknown)

Vastuse kättesaanud serveris peab olema ohjur, mis vastavalt vastusele edasi käitub (näiteks veebiserveri puhul öeldakse kasutajale, et tal on kehtetu sertifikaat ja juurdepääs edasistele teenustele on keelatud). OCSP-le seadistamise puhul tuleb Eestis AS-Sertifitseerimiskeskuselt muretseda digitaalne juurdepääsutõend, mida hakkab kasutama serveris asuv autentimisprogramm, mis võtab ühendust Sertifitseerimiskeskuse OCSP serveriga. Eesti Äriregistri veebiserveris tuligi kasutada viimast lahendust, kuna see on kõige turvalisem.

3.3 Kasutaja autentimine Eesti Äriregistris

Nagu eelpool öeldud tuli Eesti Äriregistri teabesüsteemi veebikeskkonnas kasutada nimelt OCSP kehtivuskinnituse teenust, mitte CRL failidel põhinevat. Sertifikaatite kirjeldamiseks kasutatakse kaht meetodit: PEM formaat ja DER formaat. Kummalisel kombel tahavad erinevad rakendused saada sertifikaadi andmeid erinevates vormingutes. Kasutaja autentimiseks on kõigepealt tarvis võtta tema ID-kaardi pealt sertifikaat ja kontrollida selle vastavust Sertifitseerimiskeskuse serveri informatsiooniga. Sertifikaadi kätte saamiseks kasutasin autor Apache’i serveris olevaid

globaalseid ümbruskonna muutujaid, kuhu loetakse muuhulgas kasutaja andmed ja sertifikaat. Kuna Äriregistri veebikeskkond on peaaegu tervikuna andmebaasis, siis oli kohe mugav Oracle vahenditega lugeda baasi kasutaja sertifikaati, nime, isikukoodi jms. Seejärel, kui kasutaja andmed koos autentimissertifikaadiga on baasi loetud suunatakse kasutaja vastavate sisendparameetritega edasi veebilehele, mis asub juba veebiserveris ja, mis sisaldab Java uba ning see omakorda käivitab Java programmi, kus autentimine reaalselt toimubki. Java programm kasutab JdigiDoc'i teeki, kus on kirjeldatud käsud, millega toimub OCSP ühenduse loomine ja vastuse saamine. (joonis 5.)



joonis 5.

Äriregistri andmebaasis on tabel nimega sert_aut, mis hoiab kasutajate autentimissertifikaate, OCSP päringute vastuseid, veateateid, kasutajate IP-aadresse sisse logimise aegu identifikaatoreid jms. Kasutaja nime ja/või isikukoodi kasutades kontrollitakse, kas kasutaja on üldse mõne ettevõtte esindaja, kellel on volitused antud asutusega toiminguid teha.

3.4 Näidisprogramm

OCSP kehtivusteenuse näiteprogramm näeb välja järgmine:

Kõigepealt loetakse sisse vajalikud teegid, millest tähtsaim on Digidoci teek.

```
import ee.sk.digidoc.*;
import java.io.*;
import java.security.cert.X509Certificate;
import ee.sk.digidoc.factory.*;
import ee.sk.utils.*;
import java.util.*;

public class OCSP_Test {
```

Programm loeb käsurealt sisse parameetrid, millest üks on digidoci konfiguratsiooni fail koos asukohaga ja teine sertifikaadi nimi, mida kontrollitakse.

```
public static void main(String[] args)
{
        try {
```

Konsoolis väljastatakse sisseloetud parameetrid.

```
        System.out.println
        ("Konfiguratsioonifail: " +        args[0]);
        ConfigManager.init(args[0]);
```

Tehakse kaardiga login ja kontrollitakse sertifikaati.

```
            System.out.println
            ("Sertifikaadi fail: " + args[1]);
            X509Certificate cert = SignedDoc
            .readCertificate
            (new File(args[1]));
            System.out.println
            ("Deklareerime elektroonilise notari");
            NotaryFactory notFac = ConfigManager
            .instance().getNotaryFactory();
            System.out.println
            ("Kontrollin sertifikaati");
            notFac.checkCertificate(cert);

            System.out.println("Sertifikaat kehtib!");
        } catch(DigiDocException ex) {
```

Kui programmi voog peaks kukkuma DigiDocExceptionisse on sertifikaat kehtetu.

```
            System.out.println("Sertifikaat ei kehti!");
            System.err.println(ex);
```

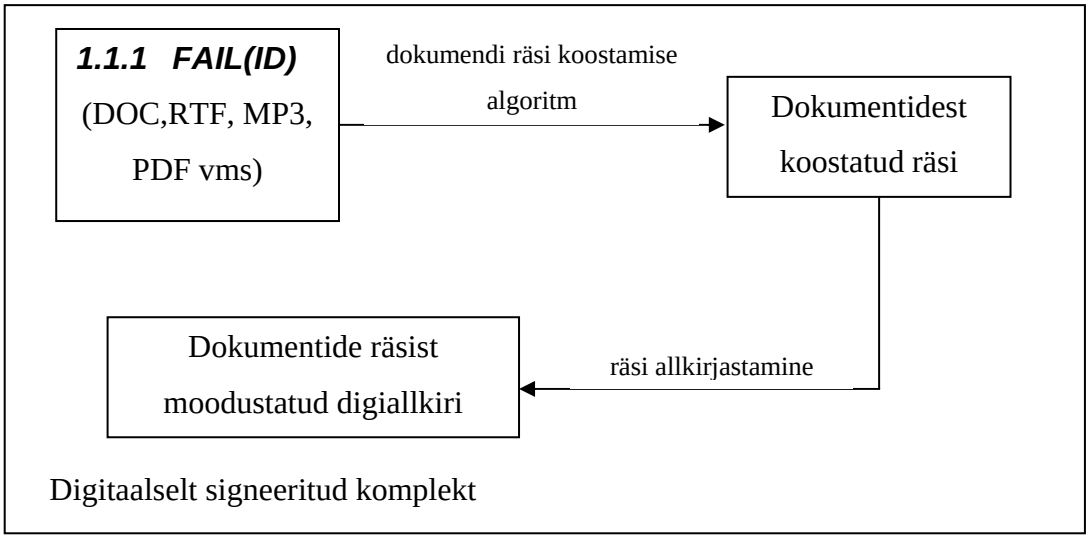
```
            ex.printStackTrace(System.err);
        } catch(Exception ex) {
            System.err.println(ex);
            ex.printStackTrace(System.err);
        }
    }
}
```

4 Digiallkirjastamine

Kui eelnevalt oli juttu autentimisest ja autentimissertifikaatidest, siis allkirjastamise jaoks on Eesti ID-kaardil teine sertifikaat – allkirjastamissertifikaat. Autentimissertifikaat põhineb samal tehnoloogial, aga sellega ei ole seotud digiallkirja seadus ja sellega ei saa isikud võtta endale kohustusi (autentimise funktsiooniga ütleb kasutaja, et ta on tema ise). Allkirjastamissertifikaadiga on seotud ID-kaardil PIN2 kood, millega aktiveeritakse signeerimisalgoritm. Eesti Äriregistri teabesüsteemile on väga oluline näha täpselt, kes on asutuse majandusaasta aruanded esitanud ja mis on tema roll firmas. Digiallkiri seob isiku dokumendiga ning pärast ei ole andmete õigsuses enam vajadust kahelda. Loomulikult võib alati tekkida situatsioone, kui ID-kaart on sattunud valedesse kätesse ja PIN-koodid on samuti teada saadud ning võib tekkida väärinformatsioon andmebaasidesse, kuid kahjuks pole selle eest keegi päriselt kaitstud.

4.1 Digitaalse allkirjastamise protsess

Tavaliselt käib allkirjastamine kohalikus arvutis. Siis kasutatakse üldjuhul DigiDoc'i programmi, mis oskab suhelda Eesti ID-kaardiga. Selle programmiga on allkirjastamine ülimalt lihtne: tuleb lohistada failid programmi aknasse, vajutada allkirjastamise nuppu ja digitaalselt allkirjastatud komplekt ongi valmis. Tehniliselt ja väga lihtsustatud kujul käib allkirjastamine järgmiselt. (joonis 6.)



joonis 6.

Enne dokumendi räsi moodustamist võetakse dokumendile värskustempel. See võimaldab allkirja andmise aega kindlaks määrata. Dokumendi räsi koostamisel on olulisel kohal ka signeerimissertifikaat (lisa 2), millest samuti sõltub räsi kuju. Võetakse dokumendile veel ajatempel ja viimaks elektroonilise notari kinnitus. Harilikult võetakse ajatempel ja notarikinnitus korraga.

4.2 Digiallkirjastamise protsess veebis

Digiallkirjade tuge saab lisada kõikidele veebis töötavatele rakendustele. Allkirja kontrollimise teostamine on suhteliselt lihtne ja seda saab teha serveri poolel oleva DigiDoci teegiga ja selle kompolentidega. Allkirjastamiseks on kliendi poolel vajalik komponent, mis suhtleks otse kasutaja ID-kaardiga, majandaks dokumentide räsidega, küsiks PIN-koode jne. Seda komponenti on võimalik teostada kahe tehnoloogia abil:

- Microsofti CAPI/CSP tehnoloogiaid kasutav komponent
- mitmeplatvormse PKCS#11 abil

PKCS#11 versioon ei ole tänaseks veel valminud.[4]

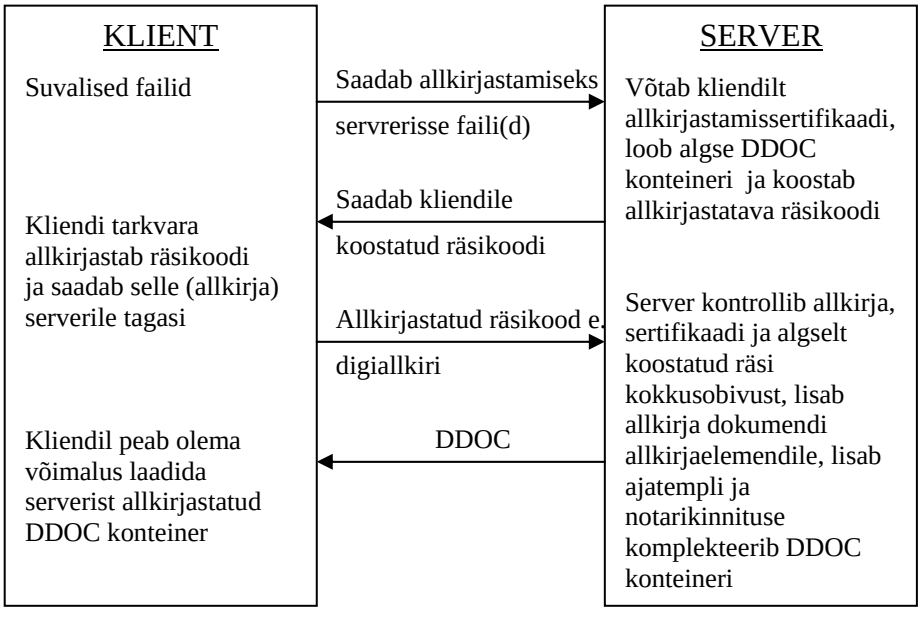
[4] http://www.id.ee

Üle veebi turvaliseks digitaalseks allkirjastamiseks on seatud mõned nõuded veebirakendustele.

- enne allkirjastamist peab kasutaja saama mõistlikul moel andmetega tutvuda, kusjuures oleks viisakas teda teavitada ka õiguslikest kohustustest, mis digiallkirja andmisega kaasnevad. Näiteks: *"**NB!** Allkirjastamise PIN-i sisestamise järel luuakse dokumendile digitaalallkiri, millest võivad allkirjastajale tuleneda õiguslikud kohustused. Seetõttu pead olema veendunud, et oled allkirjastatava info sisuga nõus. Kahtluse korral mine tagasi ja kontrolli dokumendi sisu."* [5]
- pärast allkirja andmist peab kasutajal olema võimalus kontrollida, millele tema allkiri anti, ehk saama saadida alla DigiDoci allkirjastatud komplekt koos esialgsete allkirjastatud andmete ja allkirjadega.

Muuhulgas pärinevad need punktid ka digiallkirja kohta käivast eurodirektiivist.[6]

Lihtne skeem digiallkirjastamise protsessist veebi vahendusel näeb välja järgmine: (joonis 7)



joonis 7. Digiallkirjastamise protsess

[5] http://sk.digidoc.ee

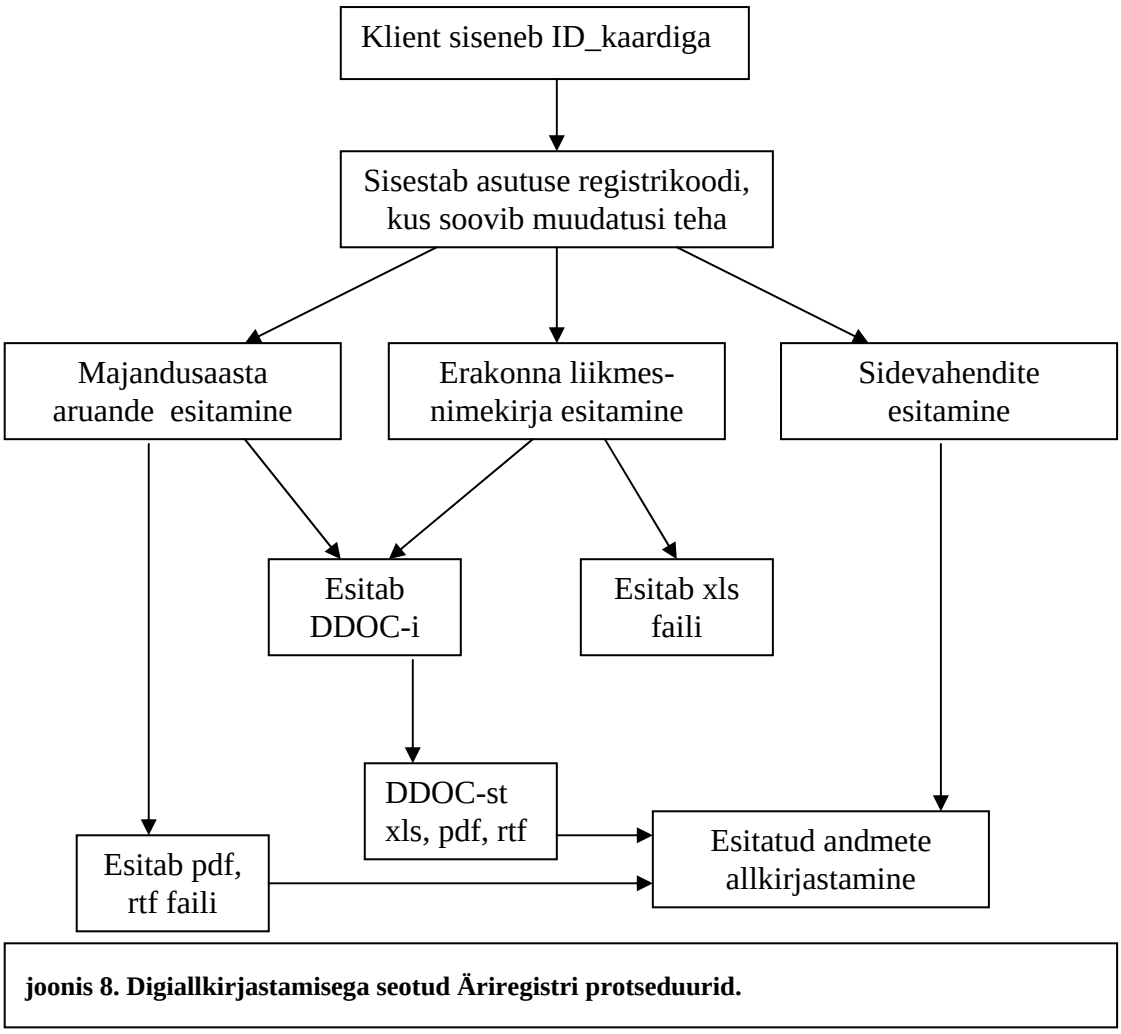
[6] http://www.id.ee

4.3 Digiallkirjastamine Eesti Äriregistri veebikeskkonnas

Äriregistri veebikeskkonnas saab esitada kolme liiki andmeid: majandusaasta aruanded, ettevõtete sidevahendite loetelud ja erakondade liikmesnimekirjad. Kui teabesüsteemi kasutaja logib end sisse ID-kaardiga siis on tal kohustus Äriregistrile esitatud andmed digitaalselt allkirjastada. Alates hetkest, kui kasutaja on end autentinud antakse talle võimalus sisestada ettevõtte registrikood, milles ta soovib muudatusi teha. Kui selgub, et tegu ei ole erakonnaga, antakse kasutajale võimalus sisestada uus majandusaasta aruanne (DDOC, RTF või PDF vormingus) või lisada ja/või muuta oma asutuse kontaktandmeid ehk sidevahendeid (kontakttelefonid, faksid, e-mailid, koduleheküljed jne).

Kui kasutaja esitab majandusaasta aruande, või erakonna nimekirja DDOC formaadis, on järelikult tema esitatud aruanne juba allkirjastatud, kuid igaks juhuks lastakse tal komplekt kinnitamisel uuesti allkirjastada. Teistkordne allkirjastamine on oluline, kuna komplektis võib olla inimeste allkirju, kellel ei ole seost antud äri- või mittetulundusühinguga. DDOC-konteineris peab olema kindlasti täpselt üks fail ja õiges ettenähtud formaadis (RTF, PDF või XLS). Seejärel võetakse DDOC failist välja majandusaasta aruande või erakonna liikmesnimekirjaga dokument pannakse see andmebaasi ja kasutaja võib teistkordselt allkirjastades uuesti kinnitada oma aruande.

Juhul, kui veebikasutaja siseneb ID-kaardiga ja laeb allkirjastamata majandusaasta aruande andmebaasi ning seejärel allkirjastab selle, jääb see ka tema lõplikuks kinnituseks. Sisuliselt moodustatakse allkirjastamisel jällegi DDOC-i konteiner, kuhu lisatakse veel XML-failiks moodustatud sidevahendite loetelu, osanike/aktsionäride nimed ja osa/aktsiakapital. Juhatuse liige samast ettevõttest saab lisada oma allkirja sellele komplektile. Allkirju eemaldada ei ole võimalik, kuid enne lõplikku kinnitamist saab tervet komplekti veel eemaldada. Skemaatiliselt on kujutatud Äriregistris digiallkirjastamisega seotud võimalused joonisel.(joonis 8)



joonis 8. Digiallkirjastamisega seotud Äriregistri protseduurid.

4.4 Digiallkirjastamise detailne protsess Eesti Äriregistri veebikeskkonnas

Laskudes sügavamale Eesti Äriregistri veebikeskkonna võimalustesse, on näha, et kõik ei ole päris nii lihtne kui esmapilgul tundub. Mis tegelikult juhtub, kui klient esitab majandusaasta aruande eelnevalt allkirjastatult ja kuidas koostatakse lõplik allkirjastatud komplekt, sellest järgnevalt veidi lähemalt koos mõnede koodinäidetega ja andmebaasi diagrammidega.

4.4.1 DDOC faili töötlemine

Äriregistri teabesüsteemi veebikeskkonda on loodud eraldi programm nimega DDOC_KONTR, mis tegeleb DDOC-failide avamisega, kontrollimisega ja baasi

salvestamisega. Sealjuures oli töökirjelduses eelduseks märgitud, et programm ei tohi salvestada ühtki faili serveri kõvakettale. Programm on kirjutatud jällegi Javas ja kasutatud on samu Java DigiDoc'i teeke. Programm aktiveerub ainult siis, kui andmebaasi salvestatava faili laiend on DDOC. DDOC failiga manipuleerimiseks on JdigiDoc teegis loodud hulk Java meetodeid. Protsess on järgmine:

Esiteks initsialiseeritakse konfiguratsioon ja tehakse Näidatakse DigiDocFactory.

```
ConfigManager.init("d://test//JDigiDoc.cfg");
DigiDocFactory digFac = ConfigManager.instance().getDigiDocFactory();
```

Seejärel luuakse ühendus andmebaasiga, kasutades Oracle JDBC draiverit:

```
Class.forName("oracle.jdbc.driver.OracleDriver”);
cn=DriverManager.getConnection("jdbc:oracle:thin:@192.168.1
.4:1521:baas","kasutaja","parool");
```

Baasist kontrollitakse kõigepealt, mis tüüpi andmed on kasutaja sinna laadinud, ehk millist teenust ta Äriregistri veebikeskkonnas kasutab (majandusaasta aruande või erakonna liikmete nimekirja esitamine)

```
if (teenus.equals("M")) tveerg = "veeb_bil_id";
//majandusaasta aruande esitamine
else if (teenus.equals("E")) tveerg = "veeb_eraknk_id";
// erakonna nimekirja esitamine
else tulemus = "-10";  //pole tegemist ei majandusaasta
aruande ega erakonna nimekirja esitamisega
```

Järgmiste käskudega võetakse baasist *veeb_bilfailide* tabelist BLOB tüüpi väljast arvatav DDOC-i fail. Muutuja *tveerg* tähistab teenust.

```
ResultSet rs=null;
PreparedStatement pstmt = cn.prepareStatement
("select blob_content,faili_tyyp,name from veeb_bilfailid
where "+tveerg+" = ? and rownum = 1");
pstmt.setInt(1,kirje_id);
rs = pstmt.executeQuery();
```

Järgmine koodosa on oluline BLOB-voo ümber konverteerimine baidimassiivi sisendvooks.

```
InputStream blobStream = blob.getBinaryStream();
InputStream sisse = new DataInputStream(blobStream);
ByteArrayOutputStream valja = new ByteArrayOutputStream();
int nr=sisse.read();
while(nr!=-1) {
        valja.write(nr);
        nr=sisse.read();
 }
byte[] massiiv=valja.toByteArray();
valjund = new DataInputStream
(new ByteArrayInputStream(massiiv));
sisse.close();
blobStream.close();
```

See on vajalik eeltöö selleks, et DigiDocFactory’t kasutama hakata.

Luuakse signeeritud dokumendi objekt ja omistatase talle eelnevalt moodustatud väljund. Seejärel kasutatakse käsku *countDataFiles()*, mis väljastab mulle DDOC failis olevate andmefailide arvu. Sobiv variant on 1, muud juhtumid püütakse kinni veatöötluse eesmärgil.

```
SignedDoc sdoc = null;
sdoc = digFac.readSignedDoc(valjund);
valjund.close();
int faile = sdoc.countDataFiles();
        if (faile == 0) tulemus = "-1"; // faile ei leitud
ddoc konteinerist
```

Seejärel kontrollin konteineri sisu, et failid oleksid õiges formaadis.

```
StringTokenizer tykid = new
StringTokenizer(df.getFileName(),".");
String tykk = "";           //siit saab faili laiendi
     while(tykid.hasMoreTokens()) {
         tykk = tykid.nextToken().toUpperCase();
     }
     if (teenus.equals("M")) {
          if (tykk.equals("DOC")) tulemus = "OK";
             else if (tykk.equals("RTF")) tulemus = "OK";
             else if (tykk.equals("PDF")) tulemus = "OK";
          else tulemus = "-12"; //pole oige laiendiga
          fail ddoc failis
          }
          else if (teenus.equals("E")) {
          if (tykk.equals("XLS")) tulemus = "OK";
                 else tulemus = "-12"; //pole oige
                 laiendiga fail ddoc failis
          }
```

```
          else tulemus = "-10";  //pole tegemist ei
majandusaasta aruande ega erakonna nimekirja esitamisega
```

Nüüd, kus failid on kontrollitud ja on selgunud, et kõik on õige, toimub vastupidine teisendus.

```
if (tulemus.equals("OK")) {
    ByteArrayOutputStream fsisu = new
ByteArrayOutputStream();
    byte[] sisu = df.getBody();
        if(sisu != null) {
                fsisu.write(sisu);
        }
        sisu = fsisu.toByteArray();
        InputStream valjund2 = new DataInputStream
        (new ByteArrayInputStream(sisu));
```

Teisendus on tehtud ja nüüd tuleb DDOC failist kätte saadud dokument uuesti baasi salvestada.

```
pstmt = cn.prepareStatement("update veeb_bilfailid set name
= replace(name,substr(name,instr(name,'/',1) +
1),?),mime_type = ?,doc_size = ?,blob_content =
empty_blob(),faili_tyyp = ? where "+tveerg+" = ? and rownum
= 1");
pstmt.setString(1,fnimi);
pstmt.setString(2,ftyyp);
pstmt.setLong(3,fsuurus);
pstmt.setString(4,tykk);
pstmt.setInt(5,kirje_id);
pstmt.execute();
ResultSet rs3 = null;
Blob blob3 = null;
pstmt = cn.prepareStatement("select blob_content from
veeb_bilfailid where "+tveerg+" = ? and rownum = 1 for
update");
pstmt.setInt(1,kirje_id);
rs3 = pstmt.executeQuery();
   if (rs3.next()) blob3 = rs3.getBlob(1);
rs3.close();

OutputStream outstream2 =
((oracle.sql.BLOB)blob3).getBinaryOutputStream();
byte[] buffer2 = new byte[10];
int loetud_baite2 = 0;
   while ((loetud_baite2 = valjund2.read(buffer2)) != -1)
   outstream2.write(buffer2,0,loetud_baite2);
outstream2.flush();
outstream2.close();
```

Sellega on kõik vastavad toimingud tehtud, selleks et kasutaja DDOC formaadis laetud fail andmebaasi jõuaks.

4.4.2 Digiallkirjastamise tehniline kirjeldus

Allkirjastamist on eelpool juba üsna põgusalt tutvustatud, seega laskutakse detailidesse ja vaadatakse, kuidas tehniliselt Eesti Äriregistris dokumentide digiallkirjastamine töötab. Sisuliselt toimub allkirjastamine kõigil kolmel juhul (majandusaasta aruanded, sidevahendite loetelu, erakondade nimekirjad) sarnaselt. Kasutatakse sama allkirjastamisprogrammi, ainult tabelid andmebaasis, mida täidetakse või uuendatakse, on erinevad. Digiallkirjastamise puhul on siinkohal juttu omaloodud allkirjastamise programmist, mis on serveri poolel. Tegelikult, aga toimub ju allkirjastamine hoopis kliendi arvutis, (nagu ka skeemilt oli näha) sest server ei saa kliendi eest allkirja anda.

Algus on programmil samasugune nagu eelmistes näidetes, kus deklareeritakse hulk muutujaid, andmebaasi ühendus ning näidatakse kätte JDigiDoci teegis asuv DigiDocFactory. Programm töötab Java Oana ja sisaldab kahte põhilist meetodit, mille poole JSP lehekülgedelt pöördutakse: räsi koostamise meetod ja lõpliku DDOC-konteineri koostamise meetod. Oluline on, et mõned DigiDocile olulised muutujad oleksid mälus sessiooni lõpuni.

Esimene oluline meetod läbi veebi allkirjastamisel on baasi laetud dokumendi ja vastava sertifikaadi räsikoodi koostamise meetod. Kõigepealt tuleb (nagu eelmises näites) andmebaasist võtta dokument ja teha see JdigiDocile arusaadavasse formaati, milleks siinjuhul on baidimassiiv. Andmebaasi päringust tulevad andmed jällegi BLOB kujul.

```
pstmt = cn.prepareStatement("select
blob_content,mime_type,name,doc_size from veeb_bilfailid
where veeb_bil_id = ? and rownum = 1");
```

BLOB-voog teisendatakse sisendvooks seejärel baidimassiivi väljundvooks ning lõpuks baidimassiiviks.

```
InputStream sisse = new DataInputStream(blobStream);
ByteArrayOutputStream malu = new
ByteArrayOutputStream();
int nr=sisse.read();
```

```
   while(nr!=-1){
   malu.write(nr);
   nr=sisse.read();
   }
byte[] massiiv=malu.toByteArray();
```

Nüüd luuakse uus signeeritud dokument, kuhu hakatakse lisama andmeid.

```
sdoc = new SignedDoc(SignedDoc.FORMAT_DIGIDOC_XML,
SignedDoc.VERSION_1_2);

 DataFile df = new DataFile();
 df.setBody(massiiv);
 df.setContentType("EMBEDDED_BASE64");
 df.setFileName(fnimi);
 df.setId("D0");
 df.setMimeType(mime);
 df.setSize(suurus);
 sdoc.addDataFile(df);
```

Kutsutakse välja andmebaasi protseduur, mis genereerib meile XML-faili, mis sisaldab lisaandmeid osanike/aktsionäride ja sidevahendite kohta.

```
cstmt = cn.prepareCall("{call tee_xml(?)}");
cstmt.execute();
```

Seejärel võetakse XML-fail baasist ASCII voona, teehakse stringiks, stringi baidimassiiviks ja lisatakse samuti DDOC-konteinerisse.

```
pstmt = cn.prepareStatement("select clob_content from
xml_clob where veeb_bil_id = ? and rownum = 1");
InputStream sisse3 = clob.getAsciiStream();
BufferedReader sisse01=new BufferedReader(new
InputStreamReader(sisse3));
String rida=sisse01.readLine();
String xmltext = null;
while(rida!=null){
  if (xmltext == null) xmltext = rida;
  else xmltext = xmltext+'\n'+rida;
  rida=sisse01.readLine();
}

DataFile df2 = new DataFile();
byte[] xmldata = ConvertUtils.str2data(xmltext);
byte[] u8b = ConvertUtils.data2utf8(xmldata,"ISO-8859-1");
df2.setBody(xmldata, "ISO-8859-1");
df2.setContentType("EMBEDDED_BASE64");
df2.setFileName("lisa.xml");
df2.setId("D1");
df2.setMimeType("text/xml");
sdoc.addDataFile(df2);
}
```

Kui komplekt on koostatud, on vaja sertifikaat andmebaasist võtta ja koostada räsikood.

```
pstmt = cn.prepareStatement("select sertif from sert_aut
where id = ? and rownum = 1");

String hexsert = rs2.getString(1);

byte[] bytesert = (hexToBytes(hexsert));
```

Võtame andmebaasist ka allkirjastaja rolli, mis lisatakse allkirjale.

```
pstmt = cn.prepareStatement("select allk_roll from sert_aut
where id = ? and rownum = 1");
```

Vastuse omistatakse stringile ja muudatakse massiiviks.

```
roll = rs.getString(1);
String[] rollimassiiv = {roll};
roll.toStringArray();
```

Järgneb selle meetodi kõige olulisem osa, kus moodustatakse räsikood.

```
SignatureFactory sigFac = ConfigManager.
                    instance().getSignatureFactory();
X509Certificate cert = SignedDoc.readCertificate(bytesert);
sig = sdoc.prepareSignature(cert, rollimassiiv, null);
byte[] sidigest = sig.calculateSignedInfoDigest();
```

Räsi muudetakse kliendi pool olevale EidCard moodulile arusaadavasse HEX formaati ning tagastatakse vastus.

```
rasi = bytesToHex(sidigest);
return rasi;
```

Teine meetod selles programmis käivitub siis, kui klient on räsikoodi allkirjastanud ja allkirja tagastanud. Allkiri (lisa 1) pannakse JSP-lehelt sert_aut tabelisse. Lõpliku komplekti sidumine allkirjaga käib järgmiselt:

```
pstmt = cn.prepareStatement("select allkiri from sert_aut
where id = ?");
pstmt.setString(1,kontr);
ResultSet rs9 = null;
rs9 = pstmt.executeQuery();
if (rs9.next()) akiri = rs9.getString(1);
rs9.close();
```

Olles allkiri kätte saanud, muundatakse see baitideks ja lisatakse signatuuri objektile.

```
byte[] sigval = (hexToBytes(akiri));
```

```
sig.setSignatureValue(sigval);
```

Seejärel hangitakse kehtivuskinnitus.

```
            sig.getConfirmation();
            cn.setAutoCommit(false);
```

Allkirjade kehtivuskinnituste kontroll.

```
            DigiDocFactory digFac = ConfigManager.
            instance().getDigiDocFactory();
            for(int i = 0; i < sdoc.countSignatures();
            i++){
                sig = sdoc.getSignature(i);
                sig.verify(sdoc, new Date(), false);
            }
```

Seejärel lisatakse allkirjastatud DDOC-fail baasi, kontrollides, kas baasis on juba selline DDOC-fail olemas, kui on, siis lisatakse ainult allkiri ja uuendatakse vastava tabeli kirjet, kui puudub selline DDOC, tehakse uus kirje ja pannakse uus komplekt baasi.

```
ResultSet rs7 = null;
pstmt = cn.prepareStatement("select blob_content from
ddoc_failid where veeb_bil_id = ? and rownum = 1");
pstmt.setInt(1,vebbilid);
rs7 = pstmt.executeQuery();
     if (!rs7.next()){  // siis tuleb lisamine

     pstmt = cn.prepareStatement("insert into ddoc_failid
     (name, blob_content, veeb_bil_id) values
     (?,empty_blob(),?)");
     pstmt.setString(1,"sk.ddoc");
     pstmt.setInt(2,vebbilid);
     pstmt.execute();
     ResultSet rs3 = null;
     Blob blob2 = null;
     pstmt = cn.prepareStatement("select blob_content from
     ddoc_failid where veeb_bil_id = ? and rownum = 1 for
     update");
     pstmt.setInt(1,vebbilid);
     rs3 = pstmt.executeQuery();
     if (rs3.next()) blob2 = rs3.getBlob(1);
     rs3.close();
     OutputStream baasivoog =
     ((oracle.sql.BLOB)blob2).getBinaryOutputStream();
     sdoc.writeToStream(baasivoog);
     baasivoog.flush();
     baasivoog.close();
     }
     else {  //tuleb olemasolevat kirjet uuendada
```

```
pstmt = cn.prepareStatement("update ddoc_failid set
blob_content = empty_blob() where veeb_bil_id = ?");
pstmt.setInt(1,vebbilid);
pstmt.execute();
ResultSet rs3 = null;
Blob blob2 = null;
pstmt = cn.prepareStatement("select blob_content from
ddoc_failid where veeb_bil_id = ? and rownum = 1 for
update");
pstmt.setInt(1,vebbilid);
rs3 = pstmt.executeQuery();
if (rs3.next()) blob2 = rs3.getBlob(1);
    rs3.close();
     OutputStream baasivoog = ((oracle.sql.BLOB)blob2)
     .getBinaryOutputStream();
     sdoc.writeToStream(baasivoog);
     baasivoog.flush();
     baasivoog.close();
}
```

Lõpetuseks püüatakse kinni programmis tekkinud vead, kui neid peaks tekkima.

```
      } catch(DigiDocException ex) {
          tulemus = "-9";
          System.err.println(ex);
          ex.printStackTrace(System.err);
      } catch(Exception ex) {
          tulemus = "10";
          System.err.println(ex);
          ex.printStackTrace(System.err);
      }
  }
```

5 Digiallkirjandusega seonduv andmebaasi pool

Kuna käesolevas töös on palju koodinäiteid, mis sisaldavad endas andmebaasi ühenduse loomist, päringute teostamist, andmebaasi andmete sisestamist ning uuendamist, on oluline ära tuua ka baasiskeem ja mõned olulisemad andmebaasi protseduurid. Siinkohal pole autor teinud väga detailset infosüsteemi kirjeldust, kuna see läheks antud projekti piiridest natuke välja. Olemite vahelised seosed on kõik üks-mitmele, kus iga grupi esimene element on keskseks ja järgnevate olemitega on tal üks-mitmele seos.

5.1 XML-lisafaili genereerimine

Põhjus, miks tekitatakse lisafail digiallkirjastatud komplektile majandusaasta aruannete ja veebist esitatud sidevahendite juurde, seisneb selles, et isik, kes oma allkirja kusagile lisab, peab olema kindel selles, et andmed, mida ta ekraanil näeb, ja aruandefail, mille ta ekraanil olevatele andmetele lisab, oleksid kindlasti vastavuses. Lisafaili formaat ei olegi tegelikult hetkel eriti oluline, see oleks võinud olla ka tavaline tekstifail. Edasise arengu huvides tehti see siiski XML-kujule (lisa 5), et andmed oleksid struktureeritud.

Programm, mis tegeleb XML-failide genereerimisega asub andmebaasis ja on Oracle baasiprotseduur, mis käivitatakse väljastpoolt allkirjakomplekti genereerimisel. Protseduuri nimi on “tee-xml”.

XML-failide genereerimiseks on loodud samuti mitmeid vahendeid, et programmeerijate tööd kergemaks muuta. Tuleb lihtsalt osata valida vastavalt oskustele ja vajadustele see õige vahend. Oracle 8i andmebaas võimaldaks järgmiste vahenditega XML-i genereerida: XMLGEN, XML-SQL for PL-SQL ja XMLDOM.

Autor eelistas kasutada XMLDOM-i võimalusi, kuna on sellega varemgi XML-i genereerinud ja autori arvates on see ka kõige paindlikum vahend XML-failide loomiseks.

XMLDOM implementeerib Dokumendi Objekti Mudeli liidest (DOM-interface), mis on heaks kiidetud W3C XML poolt.[7]

Programm ise on kirjutatud PL-SQL keeles ja põhitsükkel, mis päringutulemused XMLDOM-puusse kirjutab näeb välja järgmine:
Tsükkel loob XML-puusse sidevahendite elemendid ja annab neile väärtused.

```
for x in c1 loop

item_elmt := xmldom
     .createelement
     (doc,'Sidevahend');
sidevahend_node := xmldom
     .appendchild
     (root_node,xmldom.makenode(item_elmt));

if x.nimetus is not null then
     item_elmt :=
          xmldom.createelement
          (doc,'Liik');
     item_node := xmldom
          .appendchild
          (sidevahend_node,xmldom.makenode(item_elmt));
     item_text := xmldom
          .createtextnode
          (doc,x.nimetus);
     item_node := xmldom
          .appendchild
          (item_node,xmldom.makenode(item_text));
end if;

if x.numb is not null then
     item_elmt := xmldom
        .createelement(doc,'Number');
     item_node :=
        xmldom
        .appendchild
        (sidevahend_node,xmldom.makenode(item_elmt));
     item_text := xmldom
        .createtextnode(doc,x.numb);
```

[7] http://www.akadia.com/services/ora_gen_xml.html

```
     item_node := xmldom
        .appendchild
        (item_node,xmldom.makenode(item_text));
end if;
end loop;
```

See tsükkel paneb osanikud/aktsionärid XMLDOM-puusse koos nimedega ja aktsia/osakapitalidega.

```
for x in c2 loop
     item_elmt := xmldom
     .createelement(doc,'Osanik');
     osanik_node := xmldom
     .appendchild(root_node,xmldom.makenode(item_elmt));

     if x.eesnimi is not null then
          item_elmt := xmldom
          .createelement(doc,'Nimi');
          item_node := xmldom
          .appendchild
          (osanik_node,xmldom.makenode(item_elmt));
          item_text := xmldom
          .createtextnode(doc,x.eesnimi||' '||x.nimi);
          item_node := xmldom
          .appendchild
          (item_node,xmldom.makenode(item_text));
     else
          item_elmt := xmldom
          .createelement
          (doc,'Arinimi');
          item_node := xmldom
          .appendchild
          (osanik_node,xmldom.makenode(item_elmt));
          item_text := xmldom
          .createtextnode
          (doc,x.nimi);
          item_node := xmldom
          .appendchild
          (item_node,xmldom.makenode(item_text));
     end if;

     if x.kood is not null then
          item_elmt := xmldom
          .createelement(doc,'Kood_sunniaeg');
          item_node := xmldom
          .appendchild
          (osanik_node,xmldom.makenode(item_elmt));
          item_text := xmldom.createtextnode(doc,x.kood);
          item_node := xmldom
          .appendchild
          (item_node,xmldom.makenode(item_text));
     end if;
     if x.aadress is not null then
          item_elmt := xmldom
          .createelement(doc,'Aadress');
```

```
          item_node := xmldom
          .appendchild
          (osanik_node,xmldom.makenode(item_elmt));
          item_text := xmldom
          .createtextnode(doc,x.aadress);
          item_node := xmldom
          .appendchild
          (item_node,xmldom.makenode(item_text));
     end if;
     if x.osa_vaartus is not null then
          item_elmt := xmldom
          .createelement(doc,'Osavaartus');
          item_node := xmldom
          .appendchild
          (osanik_node,xmldom.makenode(item_elmt));
          item_text := xmldom
          .createtextnode(doc,x.osa_vaartus);
          item_node := xmldom
          .appendchild
          (item_node,xmldom.makenode(item_text));
     end if;
end loop;
```

Siin, deklareeritakse CLOB muutuja, kuhu lisatakse DOM-dokunent, mis hetkel on mälus. CLOB on Oracle andmebaasis lubatud andmetüüp ja seda saab lihtsa SQL-käsuga baasi lisada. Lõpetuseks vabastatakse mälust DOM-dokument.

```
minu_clob := empty_clob;
xmldom.writetoclob(doc,minu_clob);
xmldom.freedocument(doc);
```

Näiteks majandusaasta aruannete puhul lisatakse tabelisse *xml_clob* XML-fail järgmise SQL-käsuga:

```
insert       into       xml_clob(veeb_bil_id,clob_content)
values(nr,minu_clob);
```

5.2 Öised baasiprotseduurid

Kõik andmed, mis sisestatakse veebist, jäävad esialgu püsima Äriregistri veebibaasi. Seal on need senikaua, kuni ei ole käima läinud protseduur, mis andmed keskbaasi ja osakondade andmebaasidesse kopeerib. Need protseduurid kontrollivad veebist tulnud sidevahendite, majandusaasta aruannete ja erakondade nimekirjade olekuid. Kui olekud on sobivad (tavaliselt “L” – lõplikult kinnitatud) kopeeritakse andmed keskandmebaasi ja sealt omakorda osakondadesse. Osakondades viiakse andmetesse sisse muudatused ja

seejuures muutuvad jälle ka olekud (“P” - päevikusse kanne tehtud, “K” – korras, andmed laetud andmebaasi jms.). Oleku muudatused on olulised jällegi nendele protseduuridele, mis andmed tagasi keskbaasi kopeerivad ja mõningatel juhtudel veel keskbaasist veebibaasi kopeerivad. Kui protseduurid on midagi saatnud tagasi veebibaasi, siis võib klient näha veebist, mis tema esitatud andmetega on juhtunud (näiteks: vastuvõetud, tagasi lükatud jne.).

5.3 Majandusaasta aruande esitamine

5.3.1 Protsessi kirjeldus

Klient siseneb süsteemi, valib teenuse ja kohe tekitatakse kirje tabelisse *sert_aut*, kuhu pannakse tema autentimissertifikaat ja teenuse identifikaator, mille klient on valinud või millega tema isik on vastavusse pandud. Järgmisena sisestab klient ajavahemiku, milles soovib ta majandusaasta aruannet esitada. Seejärel laeb ta baasi, tabelisse *veeb_bilfailid*, oma majandusaasta aruande faili ja täidab ära muud kohustuslikud väljad, mis on aruande esitamisega seotud tabelis veeb_bilanss. Allkirjastades ID-kaardiga tehakse uus kirje tabelisse *sert_aut*, kuhu pannakse isiku allkirjastamissertifikaat. Seejärel genereeritakse *xml_clob* tabelisse XML-fail, kuhu lisatakse sidevahendite loetelu, tabelist *veeb_sidevah* ja osanikud/aktsionärid, kes samuti on sama, veebist esitatud majandusaasta aruande juures ning tabelis *veeb_osanikud*. See on XML-fail, mis pannakse allkirjastatud komplekti sisse (lisa 4), mis omakorda asub tabelis *ddoc_failid*, kuhu veel lisatakse tabelist *veeb_bilfailid* majandusaasta aruande fail. Allkirjastaja isikuandmed, lähevad tabelisse *veeb_allkir* (lisa 3).

5.3.2 Majandusaasta aruannete esitamisega seonduvad olemihulgad

veeb_bilanss = {veeb_bil_id, ettevotja_id, maj_algus, maj_lopp, markus, piirk, esitatud, kinnitus, olek, laadida, sisest_ikood, sisest_nimi, laadimise_viga, tyhist_selgit, reg_kp}

veeb_allkir = {veeb_bil_id, jrk, ikood, synnipaev, f_isik, kuupaev, olek, id_kaardiga, roll}

veeb_bil_failid = {name, mime_type, doc_size, content_type, blob_content, veeb_bil_id, veeb_eraknk_id, faili_tyyp, ddoc_fail, ddoc_nimi}

veeb_sidevah = {veeb_sidevah_id, veeb_bil_id, sliik, numb}

veeb_osanikud = {veeb_osan_id, veeb_bil_id, kood, nimi, eesnimi, hald_id, fyys_jur, aadress, postiind, osa_vaartus, riik}

xml_clob = {clob_content, veeb_bil_id, veeb_side_id}

ddoc_failid = {name, blob_content, veeb_bil_id, veeb_eraknk_id, veeb_side_id, allksert}

sert_aut = {id, sertif, allkiri, allk_roll, kontr, kehtivus, aeg, IP-aadress, veeb_bil_id, veeb_eraknk_id, veeb_side_id, veateade}

5.4 Erakondade liikmesnimekirjade esitamine

5.4.1 Protsessi kirjeldus

Erakondade liikmete nimekirjade esitamine toimub analoogiliselt majandusaasta aruannete esitamisele, kuna seal esitatakse samamoodi fail. Reaalses elus on erakondade nimekirju võimalik esitada kahel kombel: olemasoleva erakonna liikmete nimekirja esitamine ja asutamisel oleva erakonna liikmete nimekirja esitamine. Antud projektis on käsitletud ainult asutamisel oleva erakonna liikmete nimekirja esitamisest. XML-lisafaili siinjuures ei moodustata, kuna sidevahendeid asutamisel oleva erakonna juures ei esitata.

Isik siseneb ID-kaardiga veebikeskkonda ning valib erakonna liikmete nimekirjade esitamise. Kõigepealt luuakse tema jaoks kirje *sert_aut* tabelisse kuhu jääb tema autentimis sertifikaat, seejärel olles sisestanud andmed ja faili nimekirjaga luuakse kirje

erak_nimekirjadesse ja *veeb_bil_failidesse*, kuhu jääb tema sisestatud fail. Seejärel ta allkirjastades sisestatud faili luuakse uus kirje tabelisse *sert_aut* tema allkirjastamissertifikaadiga ja muude kohustuslike andmetega ning tekitatakse kirje *ddoc_failid* tabelisse allkirjastatud DDOC-konteineriga (Lisa 7.).

5.4.2 Erakondade liikmesnimekirjade esitamisega seotud olemihulgad

erak_nimekirjad = {veeb_eraknk_id, ettevotja_id, piirkond, esit_kpv, sisest_ikood, sisest_enimi, sisest_pnimimi, lopu_kpv, olek, laadida, laadimise_viga, arinimi, tyhist_selgit}

veeb_bil_failid = {name, mime_type, doc_size, content_type, blob_content, veeb_bil_id, veeb_eraknk_id, faili_tyyp, ddoc_fail, ddoc_nimi}

Erak_allkir = {veeb_eraknk_id, jrk, ikood, synnipaev, f_isik, kuupaev, olek, roll, id_kaardiga}

ddoc_failid = {name, blob_content, veeb_bil_id, veeb_eraknk_id, veeb_side_id, allksert}

sert_aut = {id, sertif, allkiri, allk_roll, kontr, kehtivus, aeg, IP-aadress, veeb_bil_id, veeb_eraknk_id, veeb_side_id, veateade}

5.5 Ettevõtte sidevahendite esitamine

5.5.1 Protsessi kirjeldus

Sidevahendeid sisestama asudes luuakse autentimisel *sert_aut* tabelisse kirje autentimissertifikaadiga seejärel valib isik asutuse, kus ta on juhatuse liige. Seejärel luuakse kirjed tabelisse *sert_aut*, sertifikaadi andmetega ja tabelisse *veeb_sidev* identifikaatorite ja teiste esitaja andmetega. Kliendi sisestatud sidevahendid lähevad tabelisse *v_sidev*. Allkirjastamisel genereeritakse XML-lisa, kuhu pannakse isiku

sisestatud sidevahendite loetelu ja tekitatakse kirje tabelisse *sidev_allkir* allkirjastaja andmetega. Allkirjastatud DDOC-konteiner pannakse tabelisse *ddoc_failid*.

5.5.2 Sidevahendite esitamisega seotud olemihulgad

veeb_sidev = {veeb_side_id, ettevotja_id, piirkond, esit_kpv, sisest_ikood, sisest_nimi, olek, laadida, laadimise_viga, tyhist_selgit}

sidev_allkir = {veeb_side_id, jrk, ikood, fisik, kuupaev, olek, roll, id_kaardiga}

v_sidev = {veeb_sidevah_id, veeb_side_id, sliik, numb}

xml_clob = {clob_content, veeb_bil_id, veeb_side_id}

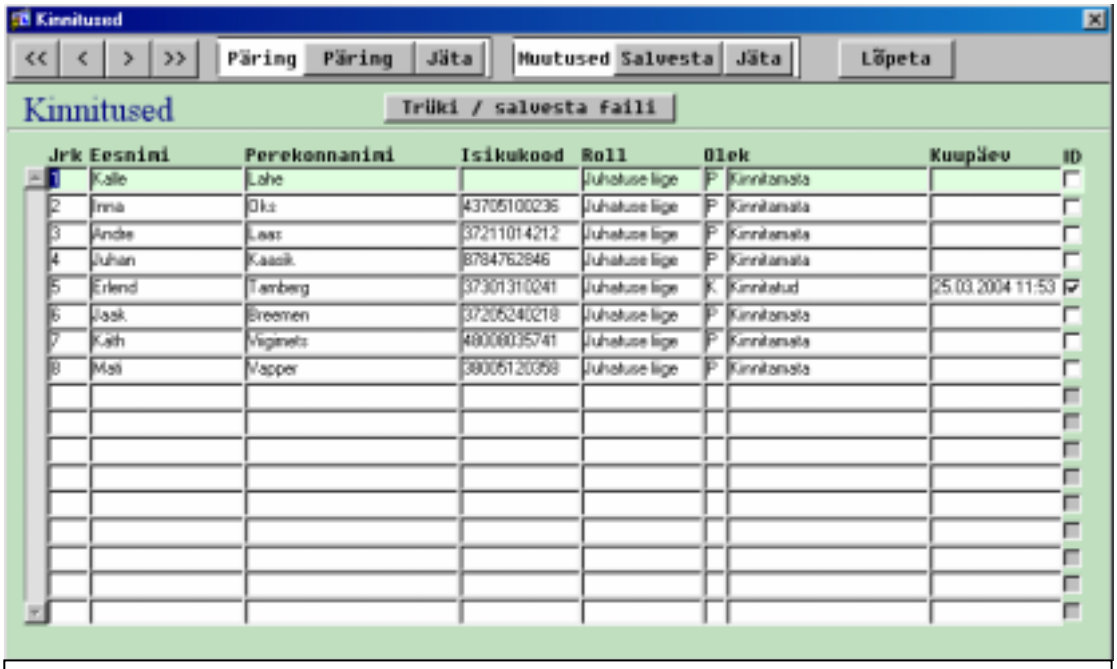
ddoc_failid = {name, blob_content, veeb_bil_id, veeb_eraknk_id, veeb_side_id, allksert}

sert_aut = {id, sertif, allkiri, allk_roll, kontr, kehtivus, aeg, IP-aadress, veeb_bil_id, veeb_eraknk_id, veeb_side_id, veateade}

6 Andmete töötlemine osakondades

6.1 Veebist esitatud majandusaasta aruanded

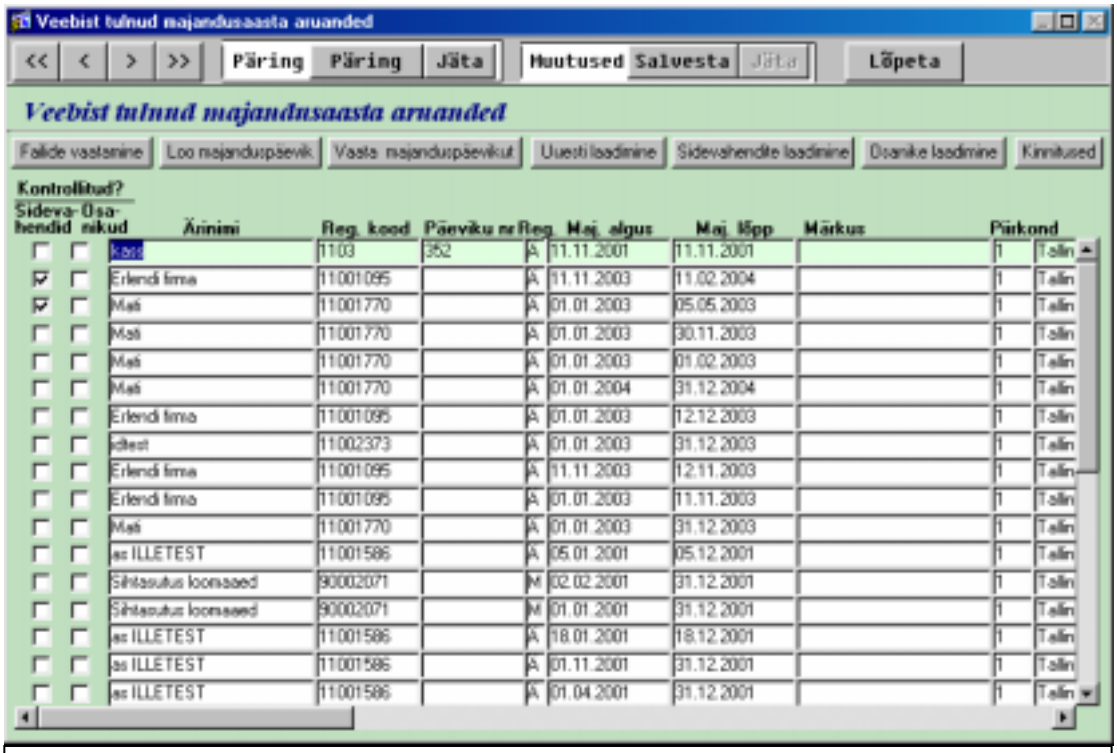
Registritöötaja näeb veebist esitatud majandusaasta aruandeid spetsiaalse ekraanivormi pealt. Vorm võimaldab töötajal avada faili, mis on aruandega kaasa pandud, kontrollida selle sisu, aruandega lisatud osanikke/aktsionäre ja sidevahendeid. Lisaks sellele on tal võimalus vaadata antud majandusaasta aruandega seotud kinnitusi ehk allkirjastajaid. (joonis 9)



joonis 9. Kinnitajate nimekiri.

Seejärel teeb ta toimingu: kas lükkab aruande tagasi või loob vastava ajavahemikuga

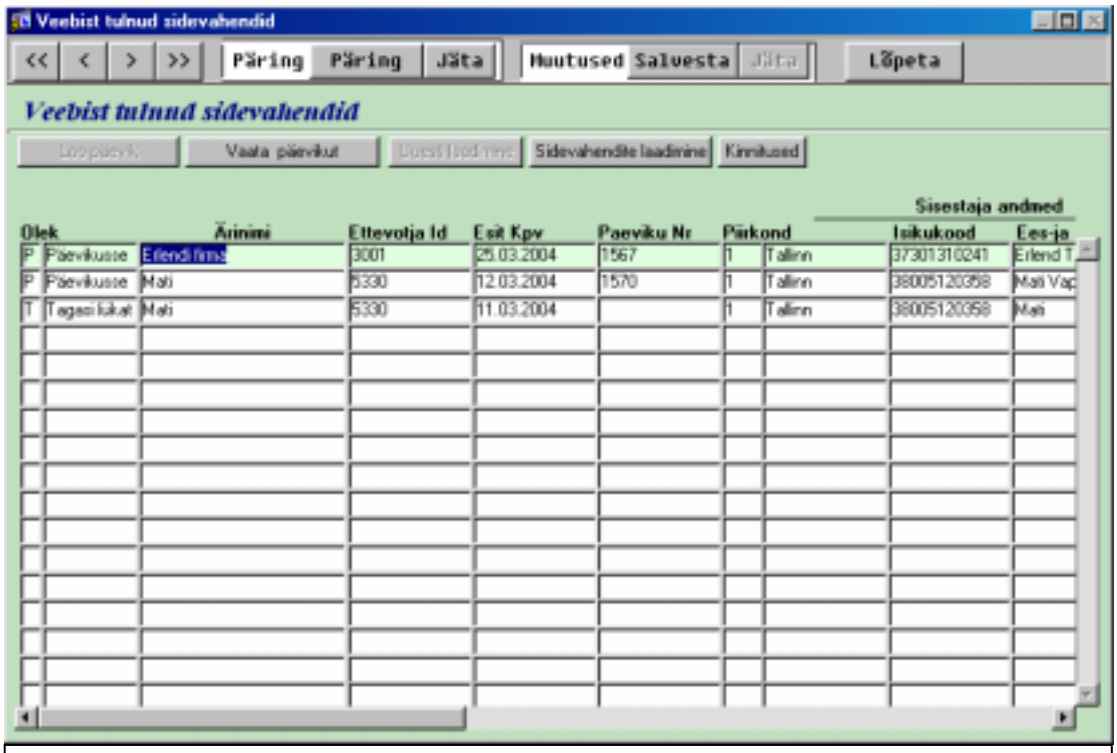
uue majanduspäeviku. (joonis 10)



joonis 10. Veebist sisestatud majandusaasta aruanded.

6.2 Veebist esitatud sidevahendid

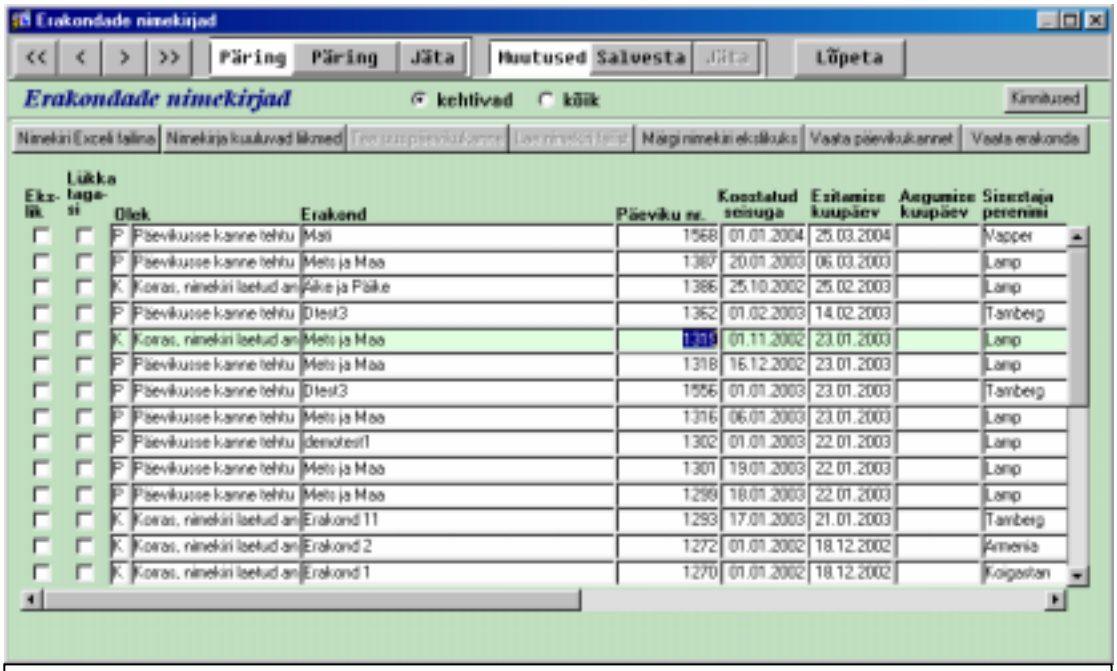
Sidevahendeid saab registritöötaja samuti sirvida ja modifitseerida vastava ekraanivormi pealt. Ka sel juhul saab vaadata sidevahendite allkirjastajaid ja kontrollida sidevahendite formaadi õigsust ning viia sisse ettenähtud muudatused. Seejärel kui andmed on korrastatud ja sobilikud, registreerib registri töötaja antud asutusega seotud uue päevikukande, millega ta lisab lõplikult esitatud sidevahendid asutuse sidevahendite loetellu. Tal on samuti võimalus need andmed tagasi lükata. (joonis 11)



joonis 11. Veebist esitatud sidevahendid.

6.3 Veebist esitatud erakondade nimekirjad

Erakondade nimekirjade esitamisel on võimalused üsna sarnased eelnevate kategooriatega. Veebist sisestatud liikmesnimekirja andmed laetakse andmebaasis olevast Exceli failist otse ekraanivormile, kus taas on võimalik teha muudatusi ja salvestada nimekiri andmebaasi. Seejuures tuleb taas luua uus päevikukanne, millega seotakse erakonna liikmesnimekiri erakonnaga. Sellelt ekraanivormilt saab vaadata ka kinnitajaid/allkirjastajaid, liikmete nimekirja; päevikukandega moodustatud nimekirju saab tagasi lükata ja veebist sisestatud nimekirju ekslikuks muuta. (joonis 12)



joonis 12. Veebist esitatud erakondade nimekirjad.

7 Kokkuvõte

Usun, et tehtud töö on suureks abimeheks igale veebiprogrammeerijale, kes oma veebilehele tahab lisada ID-kaardi tuge. Seoses suhteliselt kergele keelekasutusele peaks antud töö olema loetav, mitte ainult IT sektori inimestele, vaid pea kõikidele, kes on vähegi huvitatud Eesti ID-kaardi tööpõhimõtetest ja võimalustest. See töö peaks ka tavaimesele tegema üsna selgeks ID-kaardi ja digiallkirja süsteemi. Töös väljatoodud skeemide ja näidete varal peaks olema lihtne ka algajal veebiprogrammeerijal luua rakendusi mis toetavad Eesti ID-kaarti. Välja on toodud enamus programmide koodist koos kirjeldustega, mida on võimalik otseselt kasutada uutes loodavates rakendustes. Eesti Äriregistri liidesena on antud süsteem tööle pandud, mis küll hetkel töötab Justiitsministeeriumi registrikeskuse õppebaasis, aga peaks iga hetk hakkama toimima ka tõelises Eesti Äriregistri infosüsteemis.

8 Summary

Connecting Estonian Business Directory with Estonian ID-card system is a project which involves ID card technology based on the Public Key Interface (PKI), digital signature and authenticating technologies. The focus is set on authenticating system based on Online Certificate Status Protocol (OCSP), digital signing and digital signing based over web processes.

The chapters of the BA thesis have been arranged in the order of the actual work processes, which have been divided into introductory subchapters and subchapters which describe Estonian Business Directory's situation.

The author believes that the project's aim is to help web programmers who intend to add ID card system support to their web pages. Due to the fact that the language which has been used in the theses is relatively simple, it should be understandable not only to the people involved in IT sector, but to everyone else interested in the possibilities and usage of the Estonian ID-card system. The thesis contains examples and schemas which should provide even beginner web programmers with possibilities to implement the Estonian ID-card system in their applications. Additionally, a source code with descriptions which can be used for creating new applications has been provided in this thesis.

9 Kasutatud kirjandus

1. Valdo Praust, Digitaalallkiri. AS Kirjastus Ilo 2001.

2. Oü Allgaator (2003), konsultant Tarvi Martens. Digitaalallkiri ja ID-kaart.

3. EuroPKI Top Level Certification Authority, http://www.europki.org/ca/root/ca_cert/en_index.html

4. Akadia AG, http://www.akadia.com/services/ora_gen_xml.html

5. ID-kaardi portaal, http://www.id.ee

6. AS Sertifitseerimiskeskus, http://www.sk.ee

7. Sertifitseerimiskeskuse DigiDoc'i kasutajaportaal, https://digidoc.sk.ee

8. Jaagup Kippar, Java programmeerimise õppematerjalid. http://minitorn.tpu.ee/~jaagup/kool/java

9. Rick Greenwald and David C. Kreines, Oracle In A Nutshell. O'Reilly (2003).

10. David Flanagan, JAVA In A Nutshell. O'Reilly (2002)

11. Frank Rovitto (1992), Cooperative Development Environment

12. SQL*Plus User's Guide And Reference, version 3.1. Oracle Corporation.

13. Tom Portfolio (1992), PL/SQL User's Guide and Reference, version 2.0. Oracle Corporation.

14. Oracle (jaanuar/vebruar 2003), Ajakiri ORACLE, Simplify with Java Stored Procedures. The official magazine of Oracle Technology Network otn.oracle.com

15. Nirva Morriseau-Leroy, Martin K. Solomon, Julie Basu (2000), Oracle 8i Java Component Programming with EJB, COBRA, and JSP, Oracle Press

16. Scott Urman (2000) Oracle 8i Advanced PL/SQL Programming, Oracle Press

10 Lisad

Lisa 1. Digiallkirja näidis

Digiallkirju hoitakse Äriregistri andmebaasis HEX-kujul:

```
B3453467AE32D6A14C77315703BB8FF4C713579FBC97E23AC2733F6A8F1
B9A29715E095BC5D4D241A5A9B0FBD4AD441BFD634D12DDE463A6F363B3
63B99B6C152975D5D45EE35D52219ED1D147F0C7B5519DC6124793A6A2E
B0A5BD4E3352B36E75ED75C036E238506B56861EF1417AB1F1D593609E4
5775DE8F42C032940620
```

Lisa 2. Sertifikaatide näidised

Autentimissertifikaate hoitakse Äriregistri andmebaasis base64 kodeeringus, PEM formaadis.

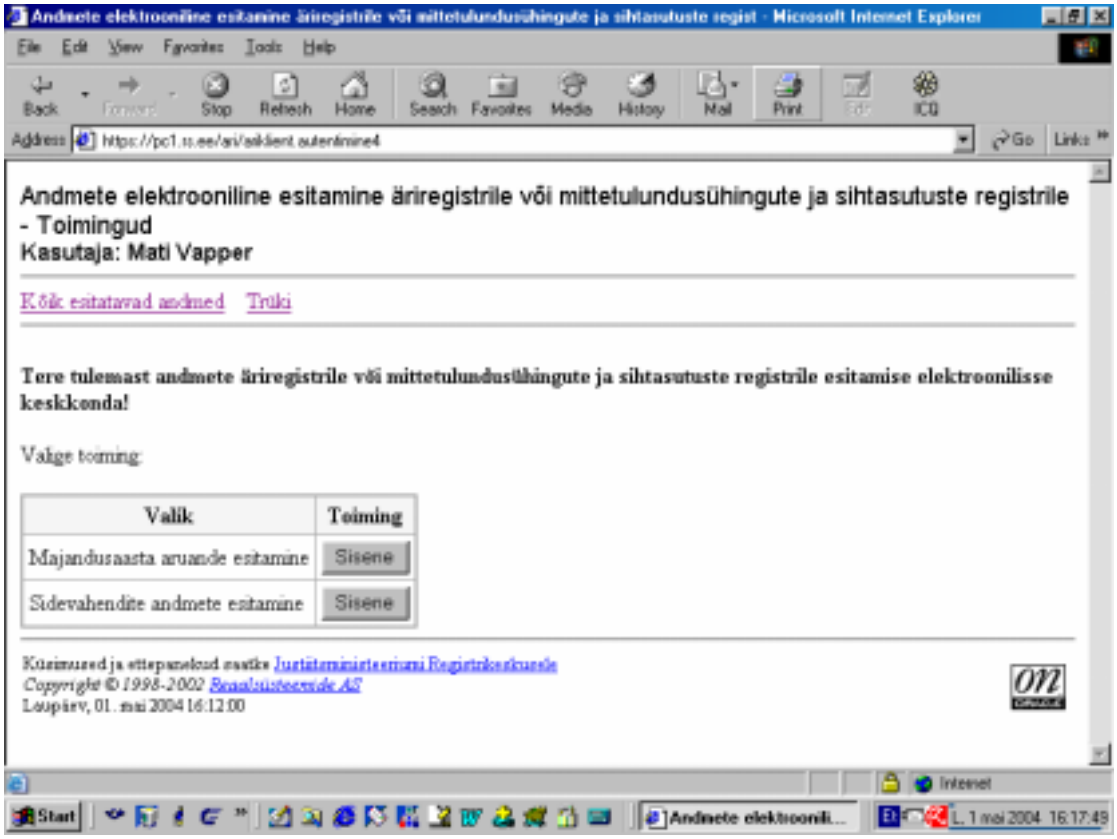

```
-----BEGIN CERTIFICATE-----
MIIEOTCCAyGgAwIBAgIEQDTcizANBgkqhkiG9w0BAQUFADB8MRgwFgYJKoZ
IhvcNAQkBFglwa2lAc2suZWUxCzAJBgNVBAYTAkVFMSIwIAYDVQQKExlBUy
BTZXJ0aWZpdHNlZXJpbWlza2Vza3VzMQ8wDQYDVQQLEwZFU1RFSUQxCjAIB
gNVBAQTATExEjAQBgNVBAMTCUVTVEVJRC1TSzAeFw0wNDAyMTkyMjAwMDBa
Fw0wNzAyMjMyMjAwMDBaMIGPMQswCQYDVQQGEwJFRTEPMA0GA1UEChMGRVN
URUlEMRcwFQYDVQQLEw5hdXRoZW50aWNhdGlvbjEgMB4GA1UEAxMXVkFQUE
VSLE1BVEksMzgwMDUxMjAzNTgxDzANBgNVBAQTBlZBUFBFUjENMAsGA1UEK
hMETUFUSTEUMBIGA1UEBRMLMzgwMDUxMjAzNTgwgZ8wDQYJKoZIhvcNAQEB
BQADgY0AMIGJAoGBAIRr9MoeJzSf4y2zF4JIkJ4HT+7aW3hNLfYhnhdN/fs
e2tNDMBK/Q2zLBUt9LIcRuxDHpChRWn+bChoAT9reI103wVxYpGEg4YBLEI
DtNcdxINDGLRtZlA828MzmLmxA7M21jRN846TbwNLpJGxoNo0iyowYSCf0k
9hgSAdcAB95AgMobgejggExMIIBLTAOBgNVHQ8BAf8EBAMCBLAwHQYDVR0l
BBYwFAYIKwYBBQUHAwIGCCsGAQUFBwMEMDgGA1UdHwQxMC8wLaAroCmGJ2h
0dHA6Ly93d3cuc2suZWUvY3Jscy9lc3RlaWQvZXN0ZWlkLmNybDAkBgNVHR
EEHTAbgRltYXRpLnZhcHBlcl80MTYzQGVlc3RpLmVlMFEGA1UdIARKMEgwR
gYLKwYBBAHOHwEBAQEwNzASBggrBgEFBQcCAjAGGgRub25lMCEGCCsGAQUF
BwIBFhVodHRwOi8vd3d3LnNrLmVlL2Nwcy8wHwYDVR0jBBgwFoAUeBe1Bfm
zWM1ZjN5nXkQGTHWGaV0w
HQYDVR0OBBYEFP7Hx3gg8hNzLJyX/clyTssK6mcsMAkGA1UdEwQCMAAwDQY
JKoZIhvcNAQEFBQADggEBAEKPn2xAckXTts2FWUxtRqz+2i+5CvXgz6RLo6
sP8NDGb0K0rl0BS40DAHHyLxYUjYfTrgbuJ9bAZE3WqY7sbPkD/qWWpgJ2k
X1KNiasrvuWuW2FMSgOCS6IKnHiO0tPBJTGDjL5M1p2Unur+fvOOgYYAKVN
pHvniGDxk/M9RmxyFhGyulaJKUu2nZJzxcJVN207swNKg+oqwPrXyvtHpXy
nfuHlEhbFwpXAVi9kvGnAVsAGD+FMojiyhY+3R2YANqCjGrq5rpWlOR/Hdn
c4hG0yRpb7QAfGd7nuM0DRHq2uVoM6Qfc7j08FJSOKLhFUxbpe8zQX/efLV
QWBVLzjHcM=
-----END CERTIFICATE-----
```

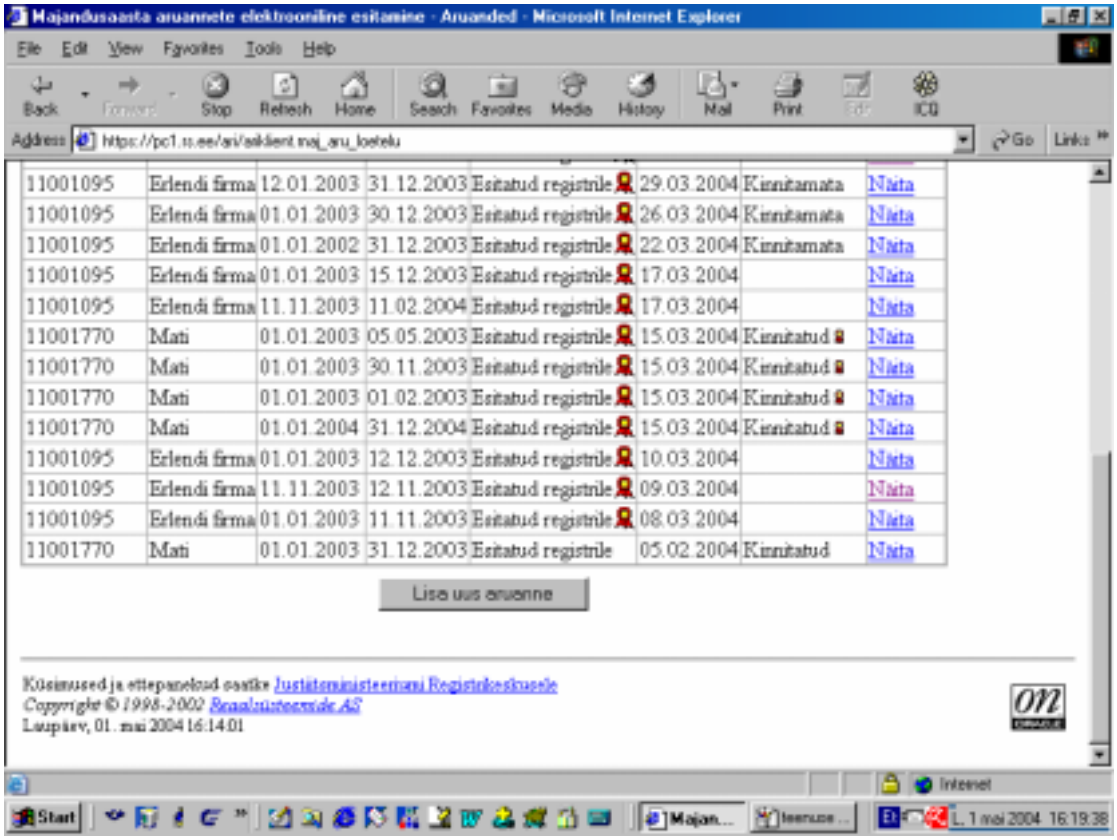

Allkirjastamissertifikaate hoitakse HEX-kodeeringus ja samuti PEM formaadis:

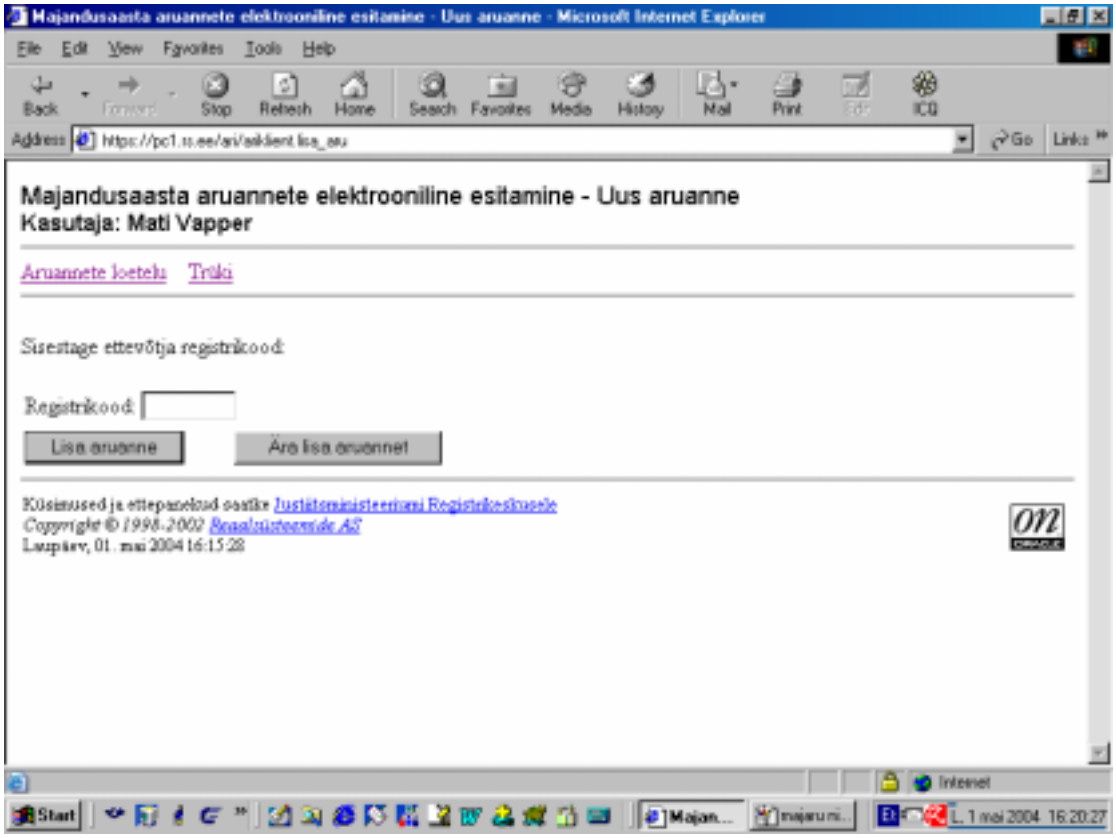

```
308203F6308202DEA00302010202044034DCA0300D06092A864886F70D0
101050500307C3118301606092A864886F70D0109011609706B6940736B
```

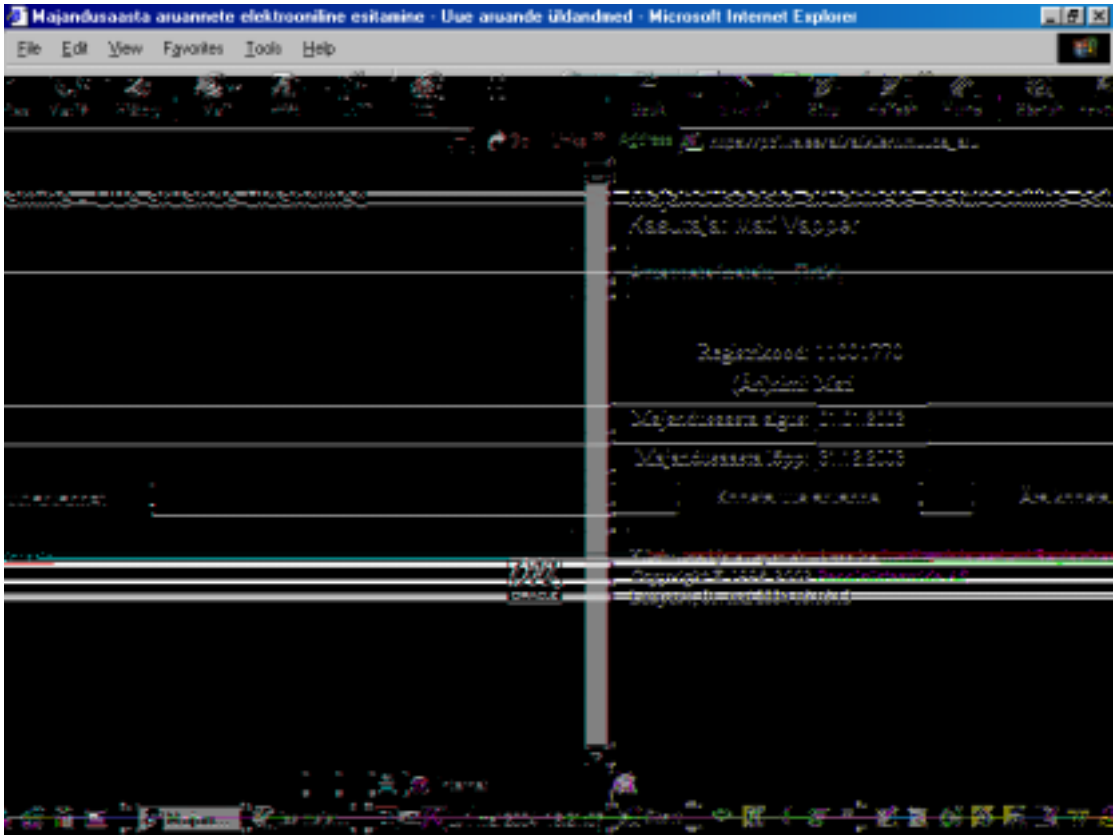
2E6565310B30090603550406130245453122302006035504 0A131941532
0536572746966697473656572696D69736B65736B7573310F300D060355
040B13064553544549443 10A3008060355040413013131123010060355 0
4031309455354454944 2D534B301E170D303430323139323230303030 5A
170D303730323233323230303030 5A308192310B3009060355040613024
545310F300D060355040A13064553544549443 11A3018060355040B1311
6469676974616C207369676E617475726531203 01E06035504031317564
1505045522C4D4154492C333830303531323033353831 0F300D06035504
041306564150504552310D300B060355042A13044D41544931143012060
3550405130B333830303531323033538308 1A0300D06092A864886F70D
010101050003818E0030818A02818100AAB130678EDD2B344226CE5E955
1557D27E813DB1A9855ED2B5857F9337A47CE9033427CC6AC7DA4ED9A06
07A889479CD72E1B3A65C3A6C5C60D6152393BED49E2833686CB6F228C9
74271197604480E2186543A67DB2F576F5767ABC66C64A6E688B6AD6AD1
F42DD60E7EA2953A3A73989CBA80B375C8FA78BBB46968B059B1020400D
8BB0FA381EB3081E8300E0603551D0F0101FF040403020640303806035 5
1D1F0431302F302DA02BA029862768747 4703A2F2F7777772E736B2E656
52F63726C732F6573746569642F6573746569642E63726C30510603551D
20044A30483046060B2B06010401CE1F0101010130373012060 82B06010
505070202300 61A046E6F6E65302106082B060105050702011615687474
703A2F2F7777772E736B2E65652F6370732F301F0603551D23041830168
0147817B505F9B358CD598CDE675E44064C7586695D301D0603551D0E04
1604140D4AAA7AE74D2A3B39857324DEBB708326080F8D30090603551D1
304023000300D06092A864886F70D0101050500038201010047B7790C64
4E75CC66A5AAF96DA621C328D406E28841D2D143FD1799670C34D125B3D
992B05F91C6CD645ADAEA160C36619410AAED0C3AD6A87BDE970B25D7EC
BB36838BD0B2A9C429D25EEF9DA1BF11025F360F6A5FB2FF42F62DDD7ED
F452999C1800C730A9950A3EEAB05E6F508137B5166D61DB729AA4BDA9E
D53C33B52CC4829402B54A1E16571083848670515A99503AF6E4A130BE9
2BCBB7501D81C6E42A285C08175FD6AAEEFB8BDB7177A32F69BA3D0B777
1906F8679BB92528239497C01C905A8E4AF3EC70EA5E80DADA3C9759705
4B2EFBAF90A769D85CC250EF0D1BB6E598E30A65E9F2884585E925F3B42
5E6AA40E817EB967FFD4C53B0BD07380

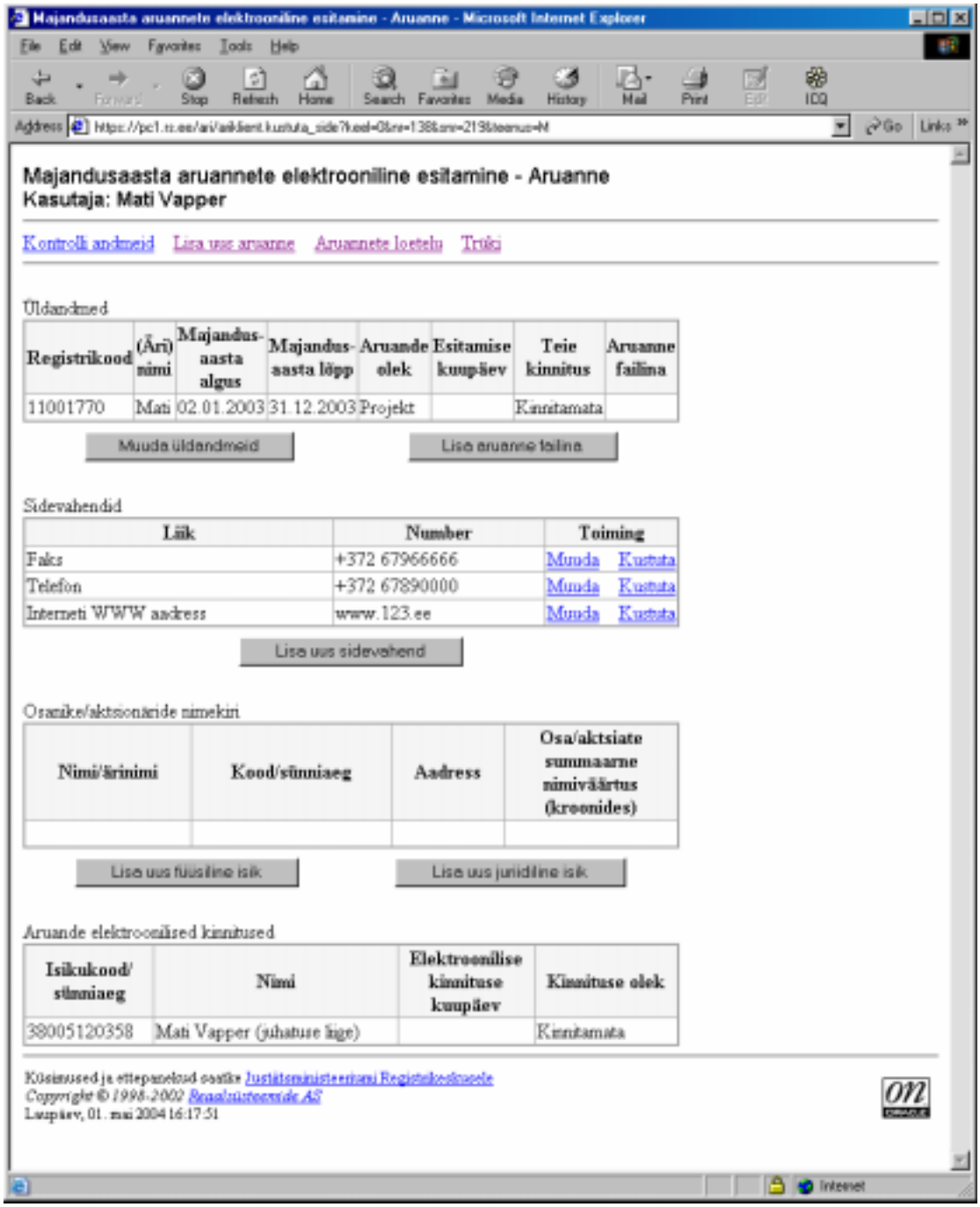
Lisa 3. Majandusaasta aruannete esitamine veebikeskkonnas

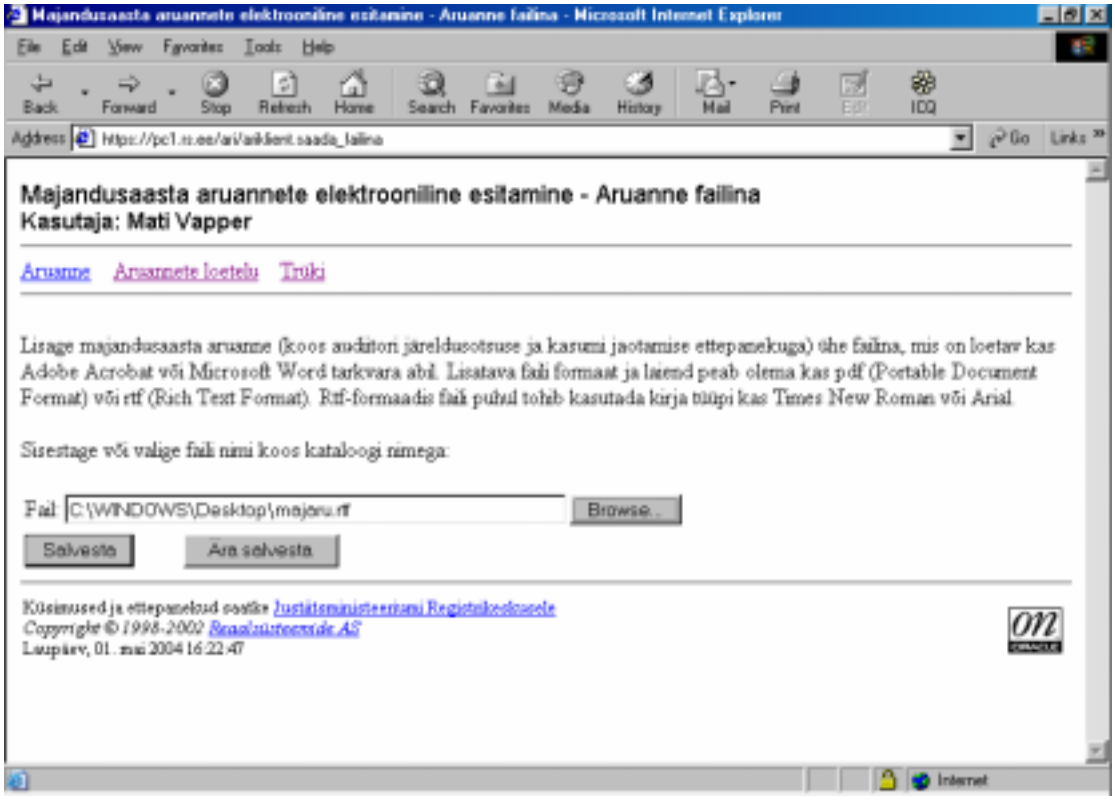


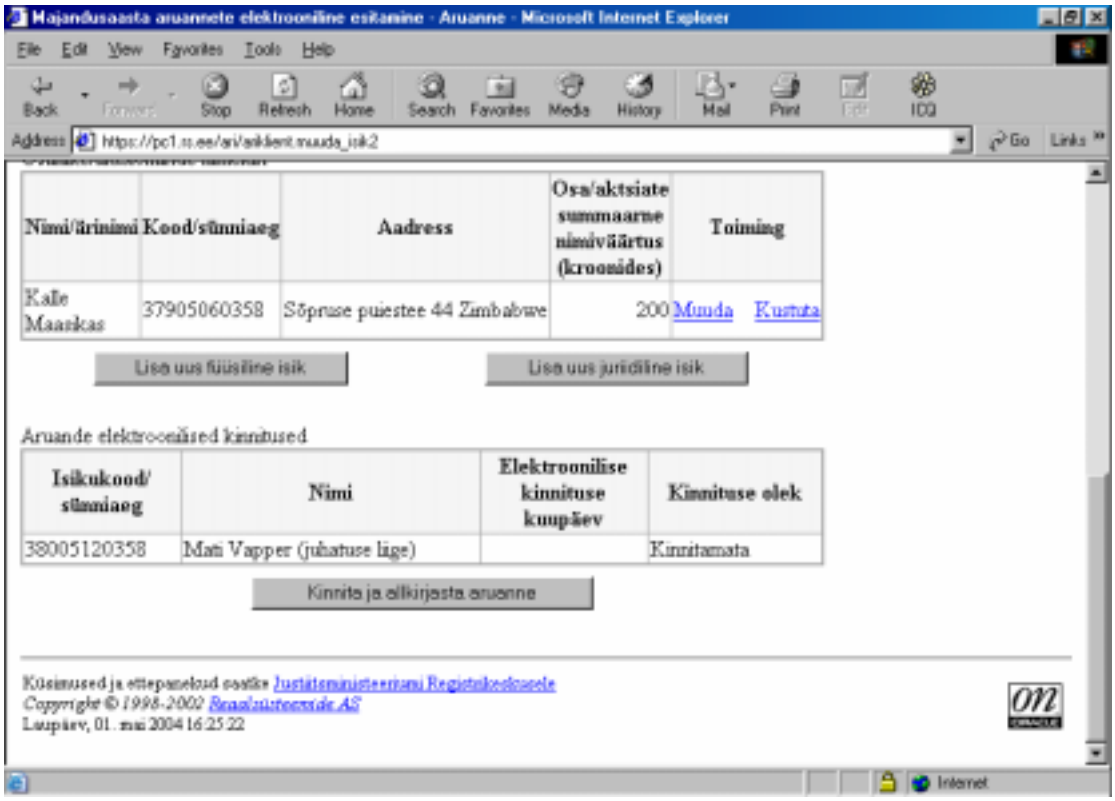


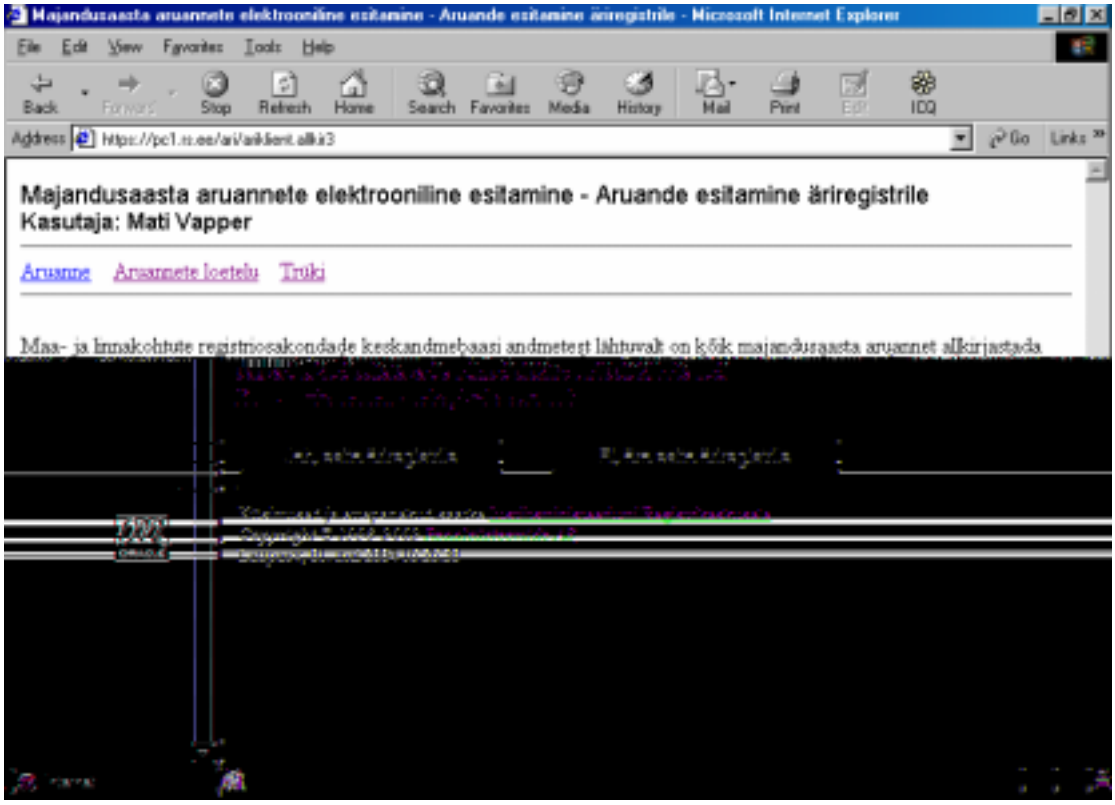

Majandusaasta aruannete elektrooniline esitamine - Uus aruanne - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Stop Refresh Home Search Favorites Media History Mail Print ICQ
Address https://pc1.ri.ee/ari/sakilent.lisa_aru
Go Links
Majandusaasta aruannete elektrooniline esitamine - Uus aruanne
Kasutaja: Mati Vapper
Aruannete loetelu Triki
Sisestage ettevõtja registrikood:
Registrikood:
Lisa aruanne
Ära lisa aruannet
Küsimused ja ettepanekud saatke Justiitsministeeriumi Registrikeskusele
Copyright © 1998-2002 Ärasõitseerimide AS
Laupäev, 01. mai 2004 16:15:28
Internet
Start
L, 1 mai 2004 16:20:27

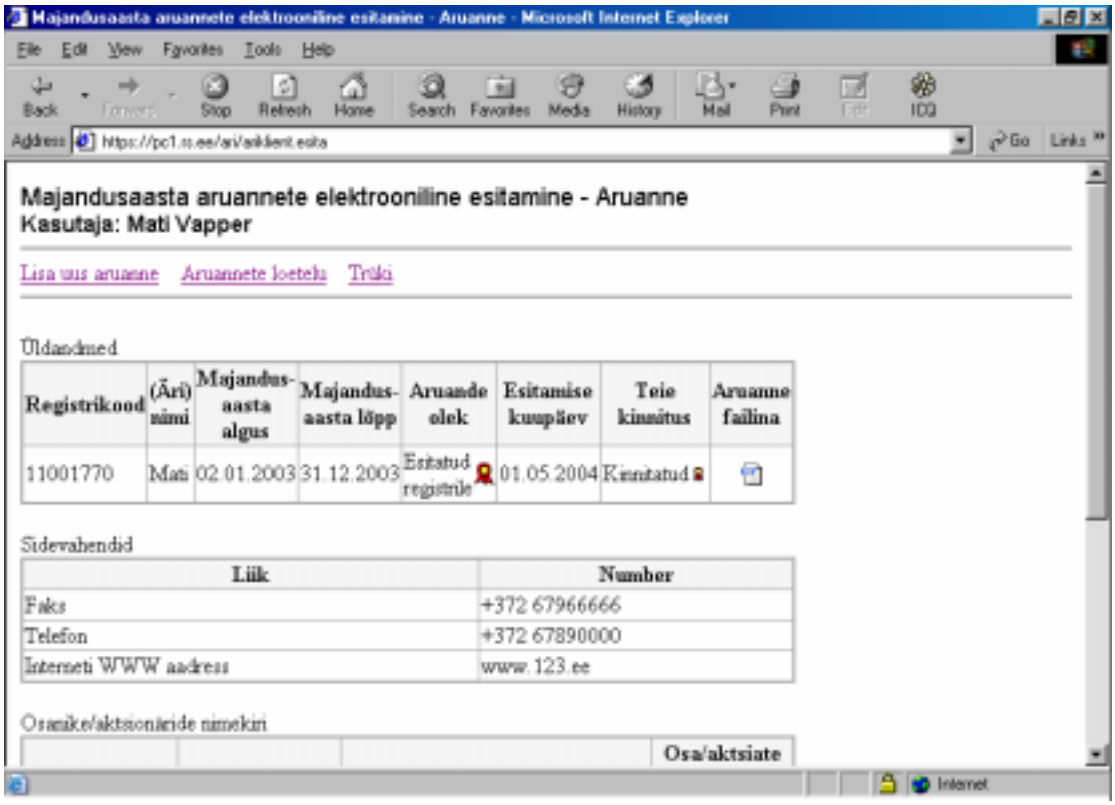

Majandusaasta aruannete elektrooniline esitamine - Uue aruande üldandmed - Microsoft Internet Explorer
File Edit View Favorites Tools Help


Majandusaasta aruannete elektrooniline esitamine - Aruanne - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites Media History Mail Print Edit ICQ
Address https://pc1.ri.ee/ari/asklient.kuntuta_side?kood=0&nr=138&snr=219&teanus=N
Go Links
Majandusaasta aruannete elektrooniline esitamine - Aruanne
Kasutaja: Mati Vapper
Kontrolli andmeid Lisa uus aruanne Aruannete loetelu Trüki
Üldandmed
Registrikood (Äri) nimi Majandus-aasta algus Majandus-aasta lõpp Aruande olek Esitamise kuupäev Teie kinnitus Aruanne failina
11001770 Mati 02.01.2003 31.12.2003 Projekt Kinnitamata
Muuda üldandmeid
Lisa aruanne failina
Sidevahendid
Liik Number Toiming
Faks +372 67966666 Muuda Kustuta
Telefon +372 67890000 Muuda Kustuta
Interneti WWW aadress www.123.ee Muuda Kustuta
Lisa uus sidevahend
Osanike/aktsionäride nimekiri
Nimi/ärinimi Kood/sünniaeg Aadress Osa/aktsiate summaarne nimiväärtus (kroonides)
Lisa uus füüsiline isik
Lisa uus juriidiline isik
Aruande elektroonilised kinnitused
Isikukood/ sünniaeg Nimi Elektroonilise kinnituse kuupäev Kinnituse olek
38005120358 Mati Vapper (juhatuse liige) Kinnitamata
Küsimused ja ettepanekud saatke Justiitsministeeriumi Registrikeskusele
Copyright © 1998-2002 Baltic Software AS
Laupäev, 01. mai 2004 16:17:51
Internet

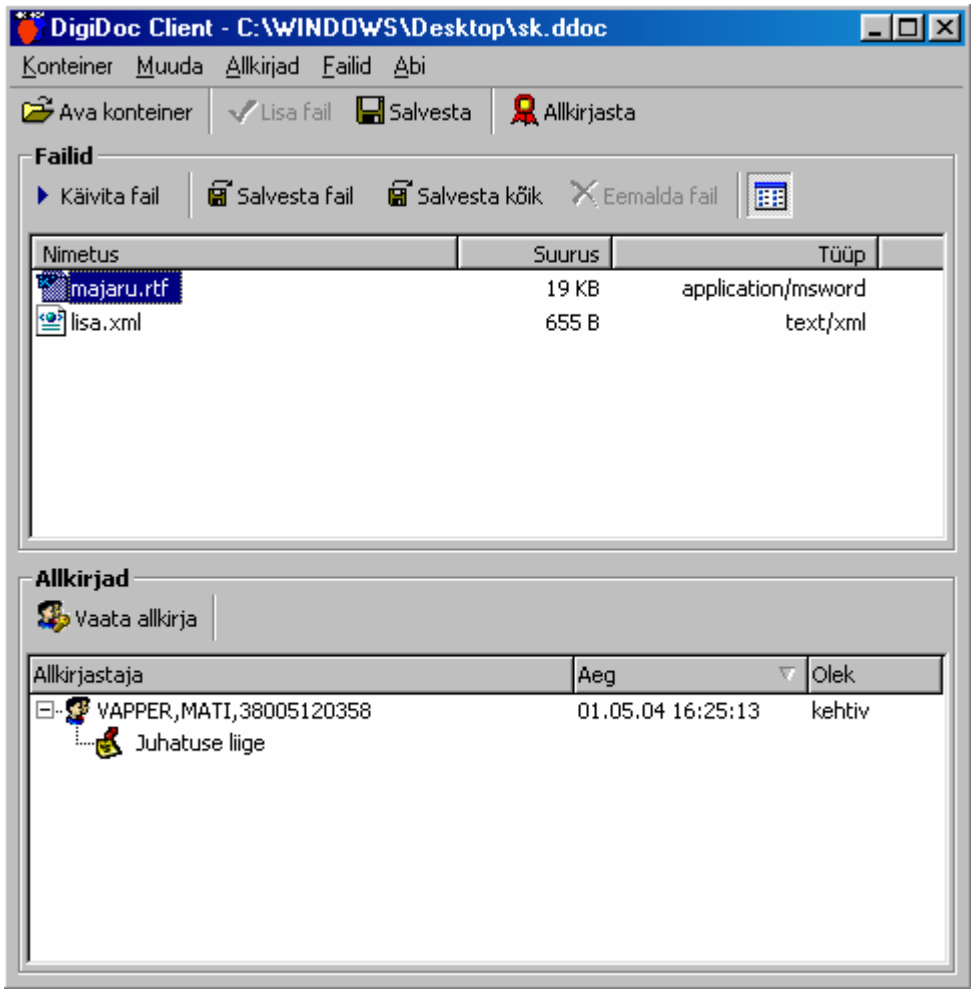

Majandusaasta aruannete elektrooniline esitamine - Aruanne failina - Microsoft Internet Explorer
Majandusaasta aruannete elektrooniline esitamine - Aruanne failina
Kasutaja: Mati Vapper
Aruanne
Aruannete loetelu
Trüki
Lisage majandusaasta aruanne (koos auditori järeldusotsuse ja kasumi jaotamise ettepanekuga) ühe failina, mis on loetav kas Adobe Acrobat või Microsoft Word tarkvara abil. Lisatava faili formaat ja laiend peab olema kas pdf (Portable Document Format) või rtf (Rich Text Format). Rtf-formaadis fail puhul tohib kasutada kirja tüüpi kas Times New Roman või Arial.
Sisestage või valige fail nimi koos kataloogi nimega:
Fail:
C:\WINDOWS\Desktop\majaru.rtf
Browse...
Salvesta
Ära salvesta
Küsimused ja ettepanekud saatke Justiitsministeeriumi Registrikeskusele
Copyright © 1998-2002 Ärasõiduteenide AS
Laupäev, 01. mai 2004 16:22:47
Internet


Majandusaasta aruannete elektrooniline esitamine - Aruanne - Microsoft Internet Explorer
Nimi/ärinimi
Kood/sünniaeg
Aadress
Osa/aktsiate summaarne nimiväärtus (kroonides)
Toiming
Kalle Maaskas
37905060358
Sõpruse puiestee 44 Zimbabwe
200
Muuda
Kustuta
Lisa uus füüsiline isik
Lisa uus juriidiline isik
Aruande elektroonilised kinnitused
Isikukood/ sünniaeg
Nimi
Elektroonilise kinnituse kuupäev
Kinnituse olek
38005120358
Mati Vapper (juhatuse liige)
Kinnitamata
Kinnita ja allkirjasta aruanne
Küsimused ja ettepanekud saatke Justiitsministeeriumi Registrikeskusele
Copyright © 1998-2002 Ärasõiduteenide AS
Laupäev, 01. mai 2004 16:25:22
Internet

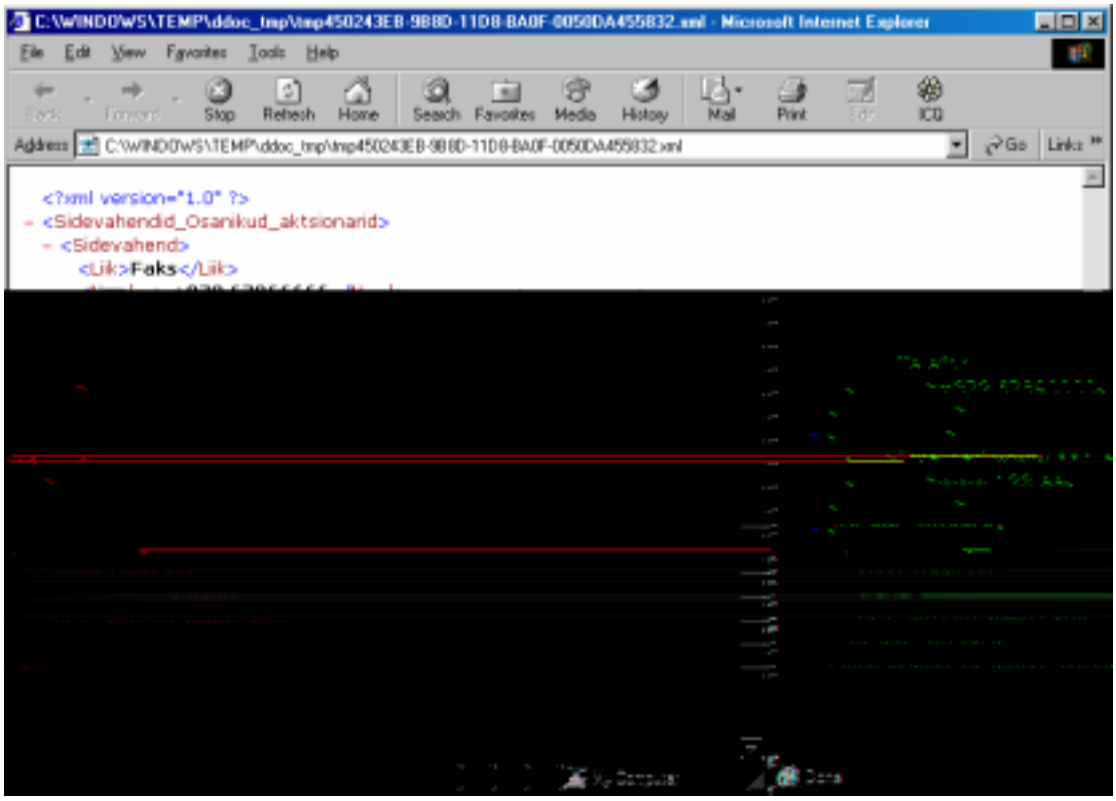

Majandusaasta aruannete elektrooniline esitamine - Aruande esitamine äriregistrile - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Stop Refresh Home Search Favorites Media History Mail Print ICQ
Address https://pc1.rr.ee/ari/aikident.alkx3
Go Links
Majandusaasta aruannete elektrooniline esitamine - Aruande esitamine äriregistrile
Kasutaja: Mati Vapper
Aruanne Aruannete loetelu Triiki
Maa- ja linnakohtute registriosakondade keskandmebaasi andmetest lähtuvalt on kõik majandusaasta aruannet allkirjastada


Majandusaasta aruannete elektrooniline esitamine - Aruanne - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Stop Refresh Home Search Favorites Media History Mail Print ICQ
Address https://pc1.rr.ee/ari/aikident.esita
Go Links
Majandusaasta aruannete elektrooniline esitamine - Aruanne
Kasutaja: Mati Vapper
Lisa uus aruanne Aruannete loetelu Triiki
Üldandmed
Registrikood (Äri) nimi Majandus-aasta algus Majandus-aasta lõpp Aruande olek Esitamise kuupäev Teie kinnitus Aruanne failina
11001770 Mati 02.01.2003 31.12.2003 Esitatud registrile 01.05.2004 Kinnitatud
Sidevahendid
Liik Number
Faks +372 67966666
Telefon +372 67890000
Interneti WWW aadress www.123.ee
Osanike/aktsionäride nimekiri
Osa/aktsiate
Internet

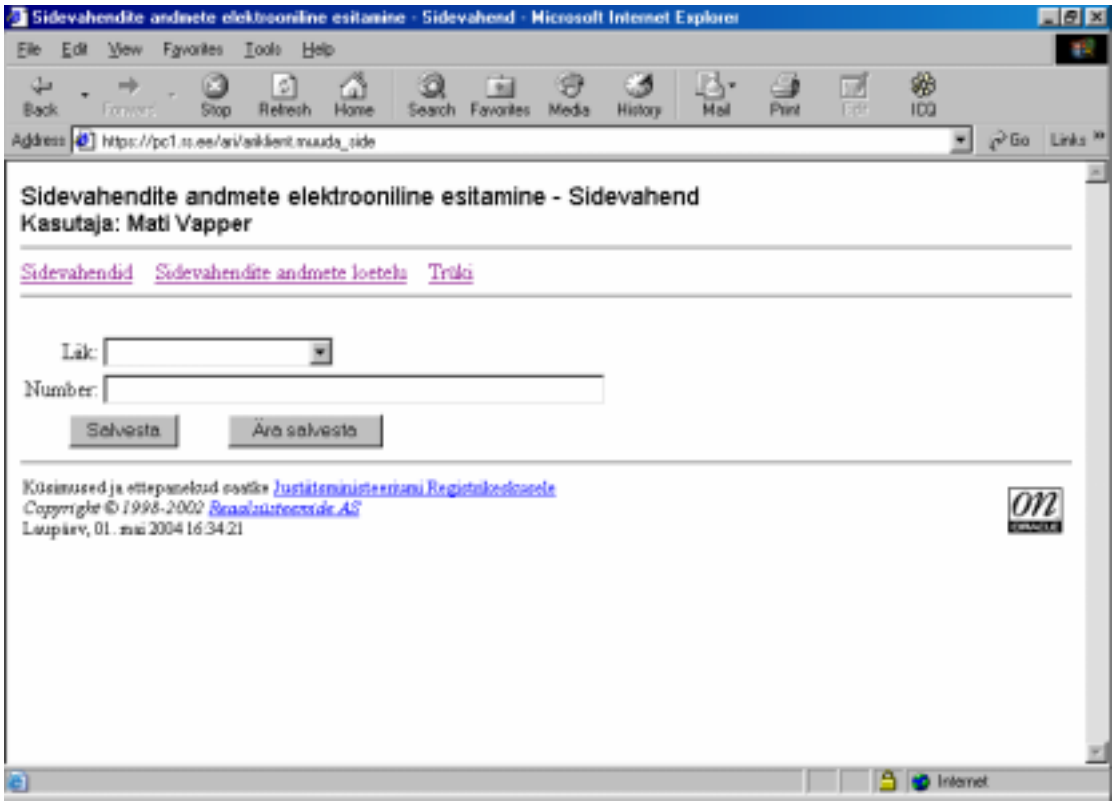
Lisa 4. Moodustatud DDOC faili näidis.

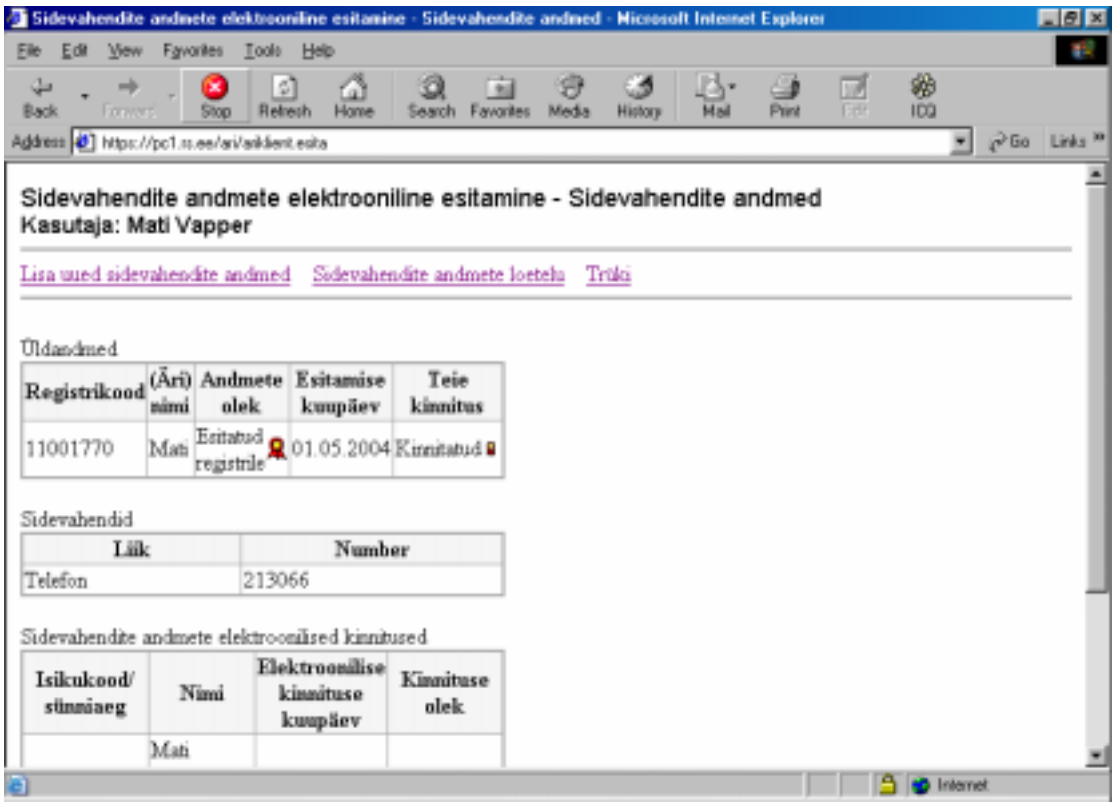


Lisa 5. Moodustatud XML-faili näidis.

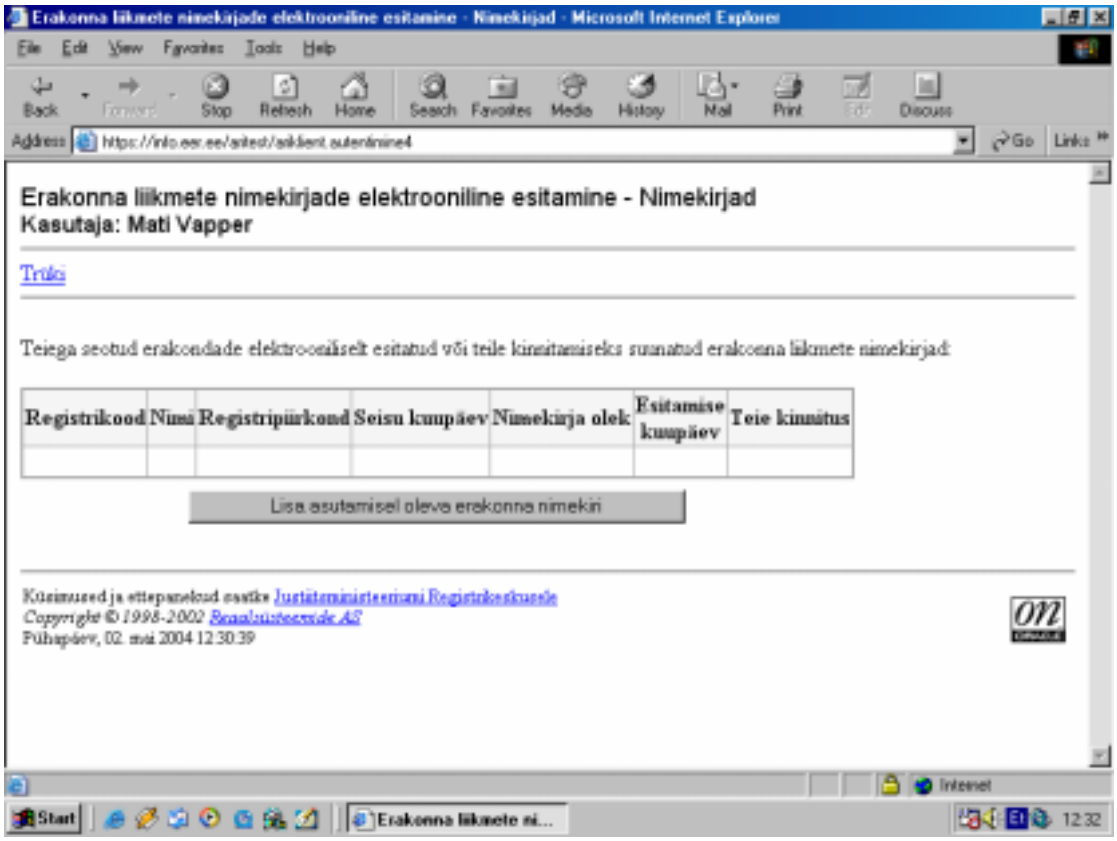


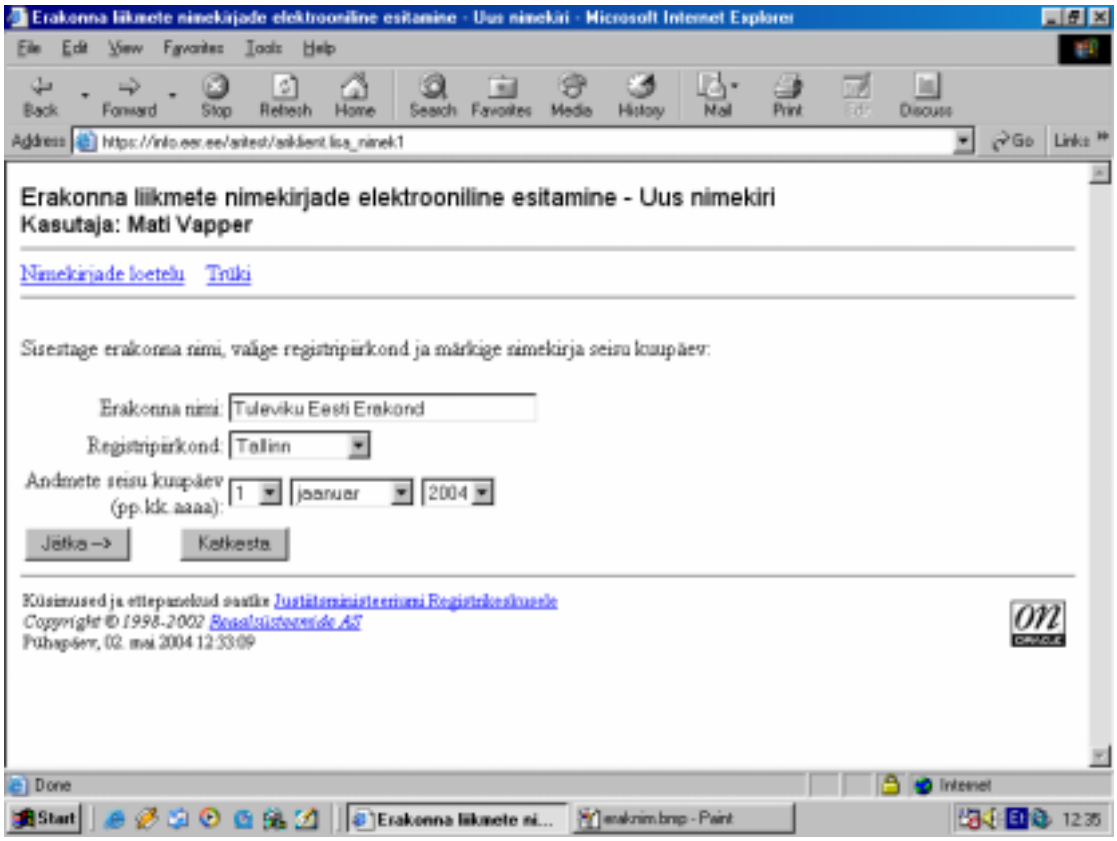
Lisa 6. Sidevahendite esitamine veebikeskkonnas.

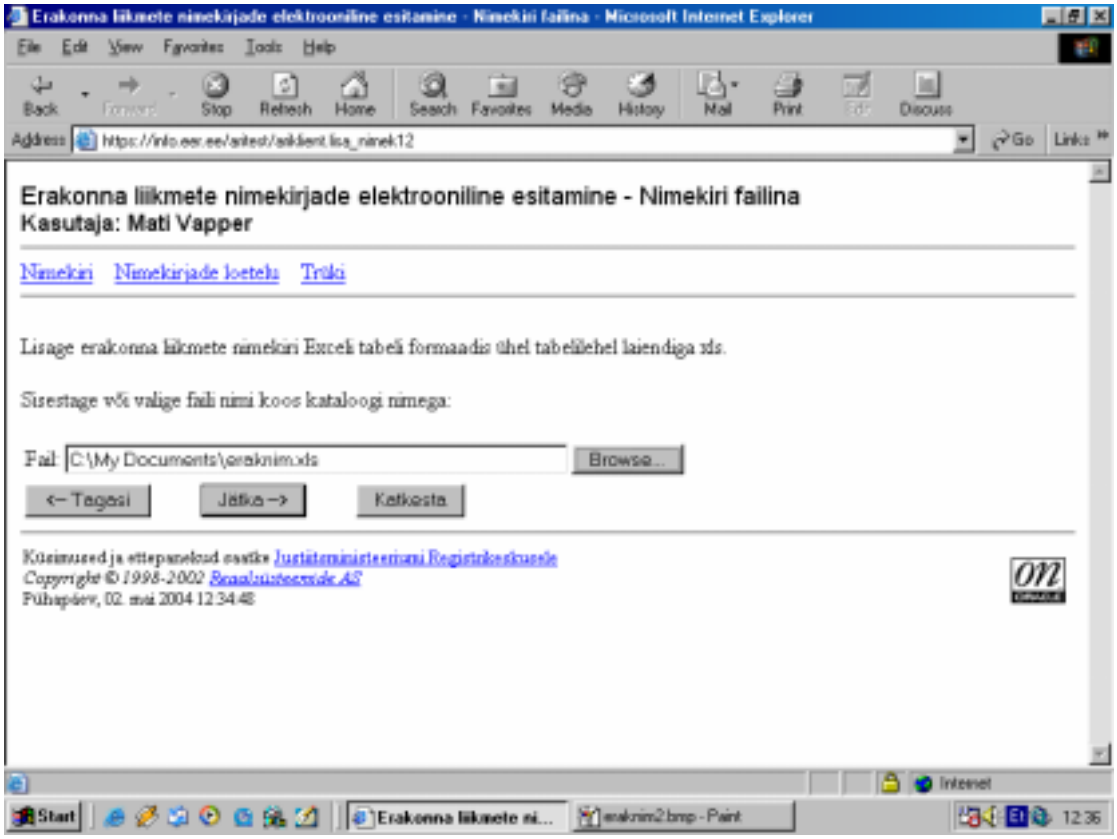


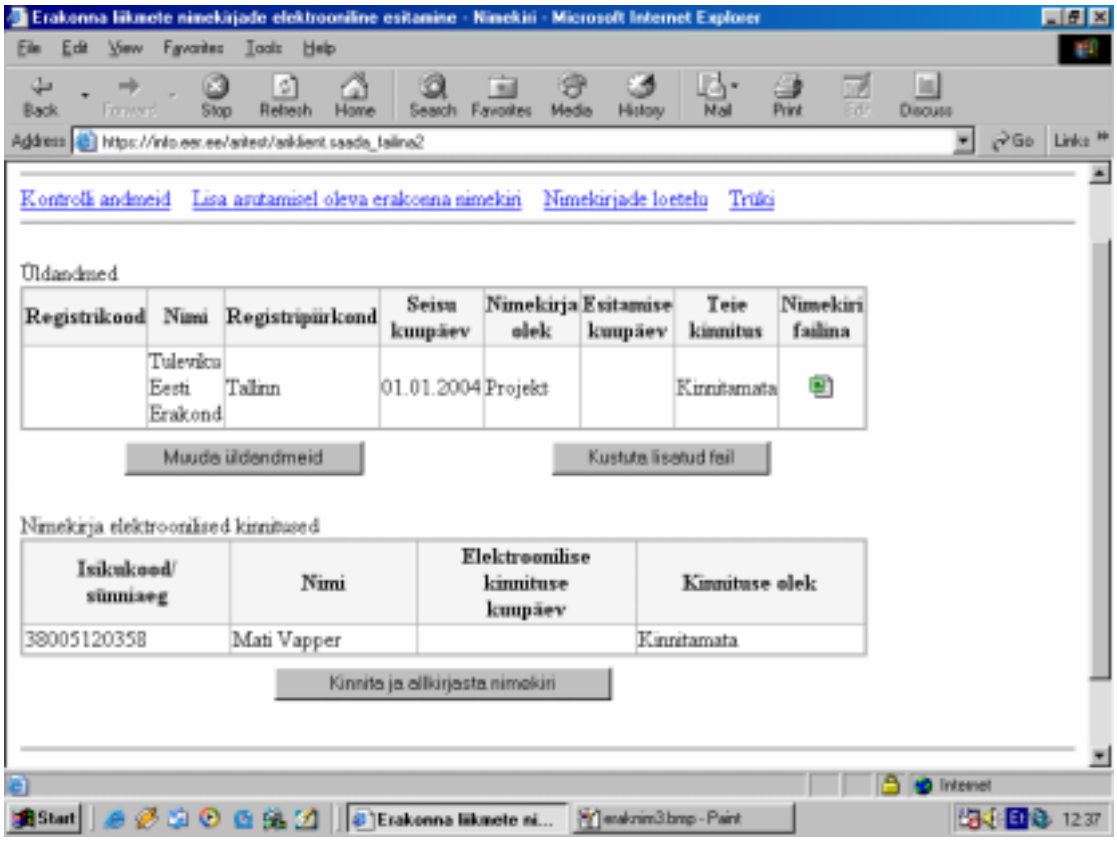


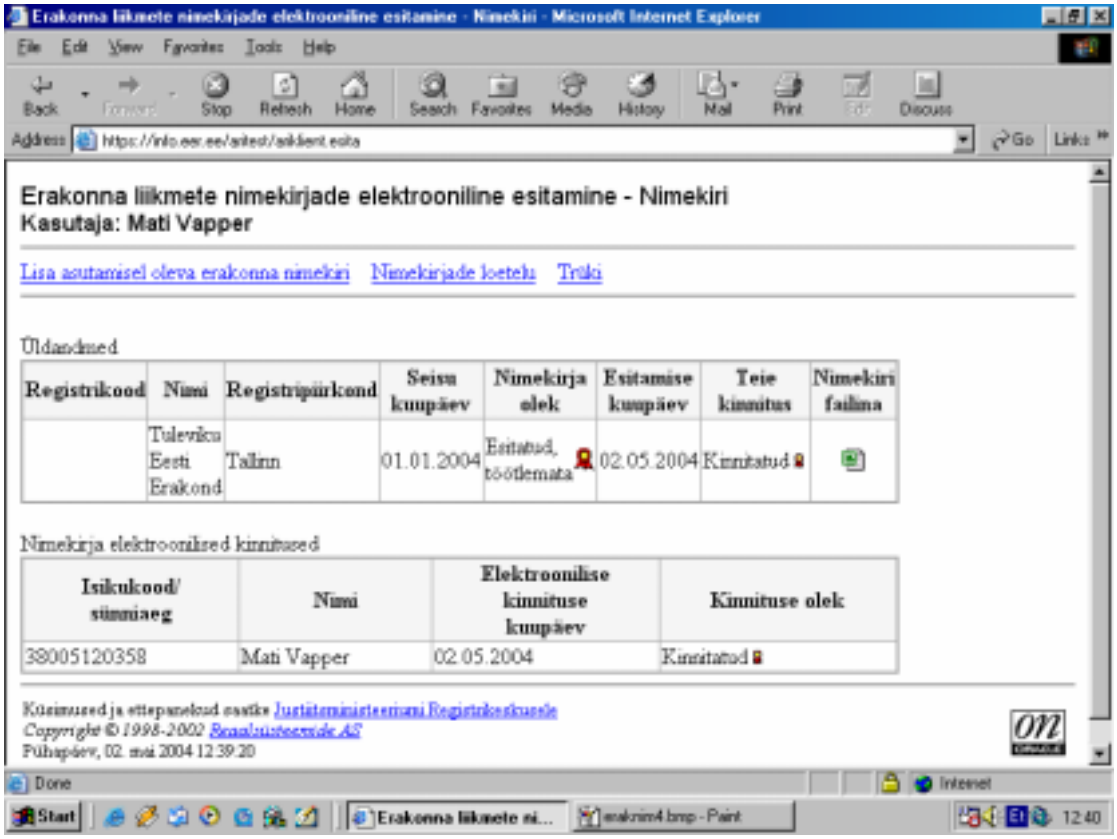
Lisa 7. Asutamisel oleva erakonna liikmete nimekirjade esitamine veebist






Erakonna liikmete nimekirjade elektrooniline esitamine - Nimekiri failina - Microsoft Internet Explorer
Erakonna liikmete nimekirjade elektrooniline esitamine - Nimekiri failina
Kasutaja: Mati Vapper
Nimekiri
Nimekirjade loetelu
Trüki
Lisage erakonna liikmete nimekiri Excel tabeli formaadis ühel tabelilehel laiendiga xls.
Sisestage või valige fail nimi koos kataloogi nimega:
Fail: C:\My Documents\eraknim.xls
Browse...
<- Tagasi
Jätka ->
Katkesta
Küsimused ja ettepanekud saatke Justiitsministeeriumi Registrikeskusele
Copyright © 1998-2002 Regatel Interactive AS
Viimati, 02. mai 2004 12:34:48


Erakonna liikmete nimekirjade elektrooniline esitamine - Nimekiri - Microsoft Internet Explorer
Kontrolli andmeid
Lisa arutamisel oleva erakonna nimekiri
Nimekirjade loetelu
Trüki
Üldandmed
Registrikood
Nimi
Registripiirkond
Seisu kuupäev
Nimekirja olek
Esitamise kuupäev
Teie kinnitus
Nimekiri failina
Tulevikku Eesti Erakond
Tallinn
01.01.2004
Projekt
Kinnitamata
Muuda üldandmeid
Kustuta lisatud fail
Nimekirja elektroonilised kinnitused
Isikukood/ sünniaeg
Nimi
Elektroonilise kinnituse kuupäev
Kinnituse olek
38005120358
Mati Vapper
Kinnitamata
Kinnita ja allkirjasta nimekiri


Erakonna liikmete nimekirjade elektrooniline esitamine - Nimekiri - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites Media History Mail Print Edit Discuss
Address https://info.ewr.ee/united/nikkent.esita
Go Links
Erakonna liikmete nimekirjade elektrooniline esitamine - Nimekiri
Kasutaja: Mati Vapper
Lisa asutamisel oleva erakonna nimekiri Nimekirjade loetelu Trüki
Üldandmed
Registrikood
Nimi
Registripiirkond
Seisu kuupäev
Nimekirja olek
Esitamise kuupäev
Teie kinnitus
Nimekiri failina
Tuleviku Eesti Erakond
Tallinn
01.01.2004
Esitatud, töötlemata
02.05.2004
Kinnitatud
Nimekirja elektroonilsed kinnitused
Isikukood/ sünniaeg
Nimi
Elektroonilise kinnituse kuupäev
Kinnituse olek
38005120358
Mati Vapper
02.05.2004
Kinnitatud
Küsimused ja ettepanekud saatke Justiitsministeeriumi Registrikeskusele
Copyright © 1998-2002 Regatel Interactive AS
Pühapäev, 02. mai 2004 12:39:20
Done
Internet
Start
Erakonna liikmete ni...
erakirm4.bmp - Paint
12:40