

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Mihkel Kivisild 174879IDDR

**X-TEE PÄRINGU VASTUSEST PDF-
FORMAADIS DOKUMENDI LOOMINE
PROKURATUURI INFOSÜSTEEMI NÄITEL**

Bakalaureusetöö

Juhendaja: Kristiina Hakk
PhD

Tallinn 2020

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: Mihkel Kivisild

12.05.2020

Annotatsioon

Lõputöö eesmärk on arendada PDF vorminguga dokumentide genereerimise loogika X-tee päringute tulemustest süsteemi halduri poolt hallatavate mallide põhjal PRIS-i (Prokuratuuri infosüsteem) näitel. Dokumentide genereerimisel tuleb arvestada kliendi poolt etteantud vormindamise nõuetega iga päringu tüübi suhtes. Töö ülesande täitmisel oli autori jaoks oluline hoida PRIS-i klientrakenduse ressursikulu võimalikult madalal ja vähendada rahalisi kulusi kliendi jaoks, uurides eesmärgi saavutamiseks vabavaralisi vahendeid. Põhilised probleemid lahenduse väljatöötamisel seisnesid korrektse vormindamise tagamisel mitme leheküljega dokumentidel ja hargtöötamise võimekuse tagamises. Vorminduse tagamiseks kasutas töö autor XSL (*Extensible Style Language*) malle X-tee päringu tulemuste XML-i (*Extensible Markup Language*) sisust HTML (*HyperText Markup Language*) dokumendi loomist, millest omakorda genereeritakse PDF dokument. Lõputöö tulemuseks on *WkHtmlToPdf* rakendusel põhinev PDF vorminguga dokumentide genereerimise lahendus, mis toetab mallipõhist lähenemist ja tagab vorminduse vastamise etteantud nõuetele.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 17 leheküljel, 7 peatükki, 6 joonist.

Abstract

Generating PDF Document from X-road Query Response by Example of Prosecutors' Information System

The goal of this thesis is to develop a PDF document generation logic from X-road query results for PRIS (Prosecutors' Information System). The resulting documents must follow the styling requirements provided by the client: generated documents must have predefined margins on each side, there must be no unreasonable gaps between elements within the document, table headers must be presented again should the table extend to a new page, page breaks must not happen between lines or table rows, and documents must have footers providing page numbering. Each X-road query type, for which the client has need for PDF generation, must have its own template. Templates must be manageable by the system administrators. The author of this thesis made his choices regarding the approach to the problem with the following principles: the solution should not bring about any licensing fees for the client and the document generation should not be reliant on end-user's system resources. With these in mind, the author sought for a freeware solution to be used on the server-side application of PRIS.

Most of the problems encountered during the development of the solution were related to ensuring the correct formatting and styling of the documents and ensuring that the generation works on a separate thread and allows multithreading for simultaneous document generations. To ensure the formatting and styling, the author developed a solution for transforming X-road query result XML into HTML using XSL templates for each query type. The final solution uses *WkHtmlToPdf* based code library to handle PDF document generation from HTML and XSL templates to handle XML transformation to HTML.

The thesis is in Estonian and contains 17 pages of text, 7 chapters, 6 figures.

Lühendite ja mõistete sõnastik

| | |
|----------------|---|
| .Net raamistik | <i>Microsoft</i> 'i poolt arendatud tarkvara raamistik, mis töötab peamiselt <i>Microsoft Windows</i> operatsioonisüsteemil |
| C# | <i>Microsoft</i> 'i poolt arendatud objekt-orienteeritud programmeerimiskeel |
| CSS | <i>Cascading Style Sheet</i> , kaskaadlaadistik |
| DLL | <i>Dynamic Link Library</i> , dünaamiliselt lingitav teek |
| DPI | <i>Dots per inch</i> , punkti tolli kohta |
| HTML | <i>HyperText Markup Language</i> , hüpertekst-märgistuskeel |
| HTTP | <i>HyperText Transfer Protocol</i> , hüperteksti edastusprotokoll |
| IE11 | <i>Internet Explorer 11</i> veebilehitseja |
| IIS | <i>Internet Information Services</i> , Interneti infoteenused |
| PDF | <i>Portable Document Format</i> , porditav dokumendiformaat |
| PRIS | Prokuratuuri infosüsteem |
| SOAP | <i>Simple Object Access Protocol</i> , lihtne objektipöördusprotokoll |
| Teek | Valmiskompileeritud alamprogramm, mida programm saab kasutada |
| UTF-8 | <i>8-bit UCS/Unicode Transformation Format</i> , 8-bitine UCS/Unicode teisendusvorming |
| WkHtmlToPdf | Vabavaraline käsurea töörist HTML dokumendist PDF dokumendi genereerimiseks |
| XML | <i>Extensible Markup Language</i> , laiendatav märgistuskeel |
| XSL | <i>Extensible Style Language</i> , laiendatav laadikeel |
| X-tee | Tehniline ja organisatsiooniline keskkond, mis võimaldab turvalist ja tõestusväärtust tagavat internetipõhist andmevahetust riigiasutuste vahel ja erasektoriga |

Sisukord

| | |
|--|----|
| 1 Sissejuhatus | 8 |
| 2 Probleemi kirjeldus ja ülesande püstitus | 9 |
| 3 Võimalikud lahendused | 11 |
| 3.1 Serveripoolne lähenemisviis..... | 11 |
| 3.2 .Net raamistikus HTML dokumendist PDF-i genereerimise võimalused | 12 |
| 4 PDF dokumendi genereerimise loogika arendamine..... | 14 |
| 4.1 <i>WkHtmlToPdf</i> rakenduse kasutamine <i>Codaxy.WkHtmlToPdf</i> teegi abil | 14 |
| 4.2 <i>WkHtmlToPdf</i> rakenduse kasutamine <i>TuesPechkin</i> teegi abil..... | 16 |
| 5 X-tee päringu tulemusest HTML dokumendi loomine | 19 |
| 6 X-tee päringutele XSL mallide loomine..... | 23 |
| 7 Kokkuvõte | 24 |
| Kasutatud kirjandus | 25 |
| Lisa 1 – Ettevõtja lihtandmete päringu XSL mall | 26 |
| Lisa 2 – Kehtivate juhilubade päringu XSL mall | 31 |
| Lisa 3 – Sõiduki detailandmete päringu XSL mall | 35 |
| Lisa 4 – Arenduskeskkonnas testandmetega loodud kinnistu detailandmete päringule vastav PDF dokument..... | 41 |
| Lisa 5 – Arenduskeskkonnas testandmetega loodud sõiduki detailandmete päringule vastava PDF dokumendi esimesed 2 lehekülge..... | 42 |

Jooniste loetelu

| | |
|---|----|
| Joonis 1. PDF dokumendi genereerimise meetod. | 15 |
| Joonis 2. <i>TuesPechkin</i> 'i teeki kasutatav PDF dokumendi genereerimise loogika. | 17 |
| Joonis 3. <i>GetResponseAsXml</i> meetod päringu tulemuse XML-i serialiseerimiseks. | 19 |
| Joonis 4. <i>GetEmptyResponseAsXml</i> meetod XML-i tagastamiseks, kui X-tee päring tagastab vea..... | 20 |
| Joonis 5. <i>GetParingAsHtml</i> meetod XML-i HTML-i transformeerimiseks. | 21 |
| Joonis 6. XSL mallide üldine kaskaadlaadistik. | 23 |

1 Sissejuhatus

Lõputöö eesmärk on prokuratuuri infosüsteemi veebirakenduse näitel luua mallipõhine lahendus X-tee päringu tulemustest PDF-formaadis dokumentide genereerimiseks. Loodavas rakenduses on kasutajatel võimalik teha üle X-tee mitmeid päringuid erinevatesse registritesse. Nende päringute tulemustest peab olema võimalik genereerida PDF-formaadis dokumente. Valmis lahenduses peavad mallid olema dünaamilised, võimaldades luua nimekirju ja tabeleid nende suurust ette andmata. Mallid peavad olema süsteemi halduri poolt hallatavad.

Töö teema on aktuaalne, sest prokuratuuri infosüsteemis puudus lahendus süsteemivälistesse registritesse tehtud päringutest PDF dokumentide loomiseks. Päringu vastuse sidumiseks menetlusega on tarvis süsteemi kasutajatel saadud vastus dokumendina saata e-toimikusse. Sellest tulenevalt on rakenduses vaja funktsionaalsust, mis võimaldaks vajalikku PDF dokumenti kasutaja jaoks võimalikult lihtsalt genereerida.

Töö koosneb sisuliselt viiest osast: probleemi kirjeldus ja ülesande püstitus, võimalike X-tee päringust PDF dokumendi genereerimise lähenemisviiside võrdlus ja valik, valitud lähenemisviisi erinevate iteratsioonide kirjeldus, ülevaade X-tee päringu tulemuste ettevalmistamisest. Töö teises peatükis kirjeldatakse ära lõputöö lähtetingimused. Kolmandas peatükis analüüsitakse kõigepealt, kas dokumentide genereerimise funktsionaalsus arendatakse eelmainitud veebirakenduse klientrakendusse või serveripoolsesse rakendusse ja tehakse vastav valik. Järgmiseks võrreldakse võimalikke teede, mida saaks kasutada lõputöö eesmärgi saavutamiseks. Neljandas peatükis antakse ülevaade arenduse erinevatest iteratsioonidest, tuues esile tekkinud probleemid. Viiendas osas kirjeldatakse X-tee päringu tulemuste vormindamist mallide abil, viies tulemused HTML kujule ja kuuendas peatükis antakse ülevaade mallide loomisest, tuues välja arenduse käigus tekkinud raskused.

2 Probleemi kirjeldus ja ülesande püstitus

Andmevahetuskiht X-tee on tehniline ja organisatsiooniline keskkond, mis võimaldab turvalist ja tõestusväärtust tagavat internetipõhist andmevahetust riigiasutuste vahel ja erasektoriga. Selle moodustavad tehniline võime andmeid vahetada ja koostalitlusvõimeline ökosüsteem. X-tee päringute sooritamiseks on andmevahetuskihiga liitunud süsteemid arendanud teenused, mida saavad teised liikmed kasutada. Loodud teenuste kasutamiseks tuleb liituda X-teega, paigaldada X-tee turvaserver, sõlmida kokkulepped sobivate X-tee teenuste osutajatega ja töötada välja loogika soovitud teenuste sisendandmete loomiseks ja vastuste töötamiseks [1].

Prokuratuuri infosüsteem (edaspidi PRIS) on prokuröridele suunatud veebirakendus, mis võimaldab rakenduse kasutajatel luua uusi menetlusi, lisada menetlustesse toiminguid, teha päringuid üle X-tee e-toimikusse ja teiste riigiasutuste registritesse. PRIS-i veebirakendus on arendatud C# programmeerimiskeeles. Rakendus pakub menetlustega seotud ametnikele keskkonda, kus kasutajal on võimalik e-toimiku vahendusel saada ülevaade erinevatest menetlustest ja teostada erinevaid toiminguid ning jagada menetlustega seotud informatsiooni teiste seotud ametnikega. Nende toimingute hulka kuulub ka päringute tegemine PRIS-i välistesse registritesse X-tee vahendusel. Kui ametnikud teevad menetluse raames erinevatesse registritesse päringuid, siis selleks, et tehtud päringut siduda konkreetse menetlusega, luuakse uus toiming. Loodud toiming saadetakse PRIS-i rakendusest X-tee vahendusel e-toimikusse. Menetluse toimingute e-toimikusse saatmisel on vajalik lisada toimingule põhifail. Selle põhifaili tekitamiseks on tarvis PDF faili genereerimise funktsionaalsust, mis looks X-tee päringu andmete põhjal uue PDF dokumendi. PDF failiformaat tagab vorminduse säilimise sõltumata kasutuskeskkonnast ja aitab tagada andmete tervikluse, võimaldamata kasutajal päringu tegemise järgselt andmeid muuta.

PDF-formaadis dokumentide genereerimiseks on kliendil olemas eraldiseisev veebiteenus, mis võimaldab HTML dokumendist PDF faili loomist. Teenuse kasutamiseks tuleb teha X-tee päring, mille sisuna edastatakse HTML dokument, millest PDF dokumenti soovitakse luua. Antud teenus ei rahuldanud kliendi poolt esitatud

vorminduse nõudeid. Sellest tulenevalt on töö autoril vajadus leida alternatiivne lahendus eesmärgi saavutamiseks. Kliendi poolt pakutava lahenduse puudustest esitatakse täpsem ülevaade järgmises peatükis.

Kliendi poolt on ette antud nõuded, kuidas loodav dokument peab olema vormindatud:

- dokumendid peavad olema vormistatud püstkirjas Times New Roman või Arial tähesuurusega 10 või 12;
- ülemise ja alumise veerise laius on 12mm, vasaku veerise laius 30mm ja parem veeris 15mm;
- joondamise tõttu ei tohi tekkida põhjendamatuid sõrendusi ega lünki;
- lehed peavad olema nummerdatud;
- tabelite päised peavad lehekülje vahetusel uuel lehel korduma;
- lehevahetus ei tohi toimuda tabeli- ega tekstirea keskel.

Pärast dokumendi genereerimist peab kasutajal olema võimalik loodud dokument alla laadida ja e-toimikusse üles laadida. Lahendus peab töötama veebilehitsejates IE11 (*Internet Explorer 11*), *Mozilla Firefox*'i, *Google Chrome*'i ja *Safari* versioonides, mis olid arenduse hetkel vastavate veebilehitsejate tootjate poolt toetatud. Veebirakenduse poolelt peab loodav lahendus töötama IIS (*Internet Information Services*) veebiserveris. Lisaks peab genereeritud dokumendis olema kuvatud ka tehtud päringu unikaalne identifikaator.

3 Võimalikud lahendused

Antud ülesande lahendamisele on võimalik läheneda kahel viisil: arendada PDF dokumendi genereerimise loogika kliendipoolsesse veebileidesse või serveripoolsesse rakendusse. Käesoleva rakenduse näitel oli mõistlik kasutada serveripoolset lähenemist, sest sageli soovib kasutaja genereeritud faili e-toimikusse üles laadida. Serveripoolsel lähenemisel salvestatakse loodud dokument failihoidlasse ja kasutajale kuvatakse faili allalaadimiseks link, vähendades sellega mälu ja protsessoriaja kulu klientrakenduses. Lisaks välditakse sellega potentsiaalseid probleeme lahenduse ühilduvusega erinevate veebilehitsejatega.

3.1 Serveripoolne lähenemisviis

Kuna X-tee päringud ja vastused on SOAP (*Simple Object Access Protocol*) sõnumid ja vastavate teenuste kasutamiseks loodud adapterites on olemas XML-i serialiseerimise loogika ning dokumendi genereerimine peab toimuma rakenduse halduri poolt lisatava malli põhjal, siis sai otsustatud, et päringu tulemused konverteeritakse XSL (*Extensible Style Language*) mallide abil XML-ist (*Extensible Markup Language*) HTML (*HyperText Markup Language*) kujule. Täiendavaks põhjuseks HTML-i konverteerimiseks oli kliendi poolt varasemalt loodud X-tee teenus, mis teostab PDF dokumendi genereerimist HTML dokumendist. Esmaseks testimiseks kasutati lihtsat HTML dokumenti, milles on tekstilõigud ja tabelid erineva laadistikuga.

Esimese lahendusena katsetati kliendi poolt loodud veebiteenust, mis kasutab PDF dokumendi loomiseks *OpenOffice Writer* rakendust. Selle variandi eeliseks antud rakenduses on vähene ajakulu arendusele, sest tarvis on ainult arendada adapteri teenuse kasutamiseks ja päringu tulemusest HTML dokumendi loomise loogika. Esmastel katsetustel selgus kohe, et seda teenust kasutades ei ole võimalik tagada kliendipoolsete nõuete täitmist. Suurimateks probleemideks olid kontrolli puudumine lehekülje nummerduse üle, põhjendamatute lünkade esinemine erinevate HTML-i elementide vahel ja lehekülje vahetusel tabelirea poolitamine ja jätkamine uuel lehel. Kuna selle lahendusega ei olnud võimalik kliendi poolt seatud nõuete täidetust tagada, siis otsustas

diplomitöö autor kliendi, projektijuhi ja projekti tarkvara arhitekti nõusolekul uurida alternatiivseid võimalusi .Net raamistikus HTML dokumendist PDF dokumendi genereerimiseks.

3.2 .Net raamistikus HTML dokumendist PDF-i genereerimise võimalused

Potentsiaalsete lahendustena uuriti vabavaralisi teke ja rakendusi, soovides vältida kliendi jaoks lisakulude tekkimist. Kõige enam leidis vasteid, mis viitasid *HtmlRenderer*'i [1] ja *iTextSharp*'i [2] vabavaralise versiooni teegile ning *WkHtmlToPdf* [3] programmil baseeruvatele lahendustele. Kõigepealt uuriti *iTextSharp* ja *HtmlRenderer* teke, sest need on arendatud C# programmeerimiskeeles erinevalt *WkHtmlToPdf* programmist. Sellest tulenevalt on neid võimalik lihtsamini integreerida olemasolevasse serveripoolsesse rakendusse.

iTextSharp'i vabavaralise teegi puhul on tegemist vanema versiooniga tasulisest *iTextSharp*'i teegist. Vabavaralist versiooni ei uuendata ega toetata teeki pakkuva ettevõtte poolt. Sellest tulenevalt ei ole seda võimalik antud ülesande lahendamiseks kasutada, sest kliendi poolt sätestatud mittefunktsionaalsetes nõuetes on märgitud, et kolmanda osapoole komponentide versioonide eluea lõpp ei tohi olla arendamise hetkest teadaolevalt vähem kui kaks aastat hiljem.

HtmlRenderer'i teek vastas arenduse ajal eelmainitud mittefunktsionaalsele nõudele. Katsetamise käigus selgus, et selle teegiga esines probleeme lehekülje vahetusega ja tabelite loomisega PDF dokumendis. Näiteks tekkis loodud dokumenti juurde tühje lehti. Lisaks esines mitmeid laadistiku rakendamise probleeme ja leheküljenumbrite lisamine dünaamiliselt genereeritavasse dokumenti ei olnud arendamise hetkel kergesti saavutatav. Käesoleva teegi kasutuselevõtu vastu rääkis ka asjaolu, et sellel puudus dokumentatsioon teegi poolt pakutava funktsionaalsuse kohta ja kõiki vigu, mis kasutusel esile kerkisid, tuli iseseisvalt uurida. Sellest tulenevalt otsustati antud teeki mitte kasutada, et vältida potentsiaalsete vigade tekkimist, mille korral on raske ette hinnata arenduskulu puuduliku dokumentatsiooni tõttu.

WkHtmlToPdf on eraldiseisev konsoolirakendus, mis on arendatud C++ programmeerimiskeeles. Sellest tulenevalt ei saa seda otseselt integreerida

olemasolevasse serverirakendusse, vaid see peab olema serverisse installeeritud ja serverirakendus viitab sellele installatsioonile. Selle programmi positiivseks küljeks on dokumentatsiooni olemasolu, mis andis juba varakult teada rakenduse võimekusest. Lisaks on rakendusele sisseehitatud eraldiseisvate päiste ja jaluste lisamise loogika, mis võimaldas lehekülgede nummerdamist ilma selleks eraldi suuremat arenduskulu nõudmata. Põhjalik ülevaade rakenduse võimekusest tagas suurema enesekindluse lõputöö eesmärgi saavutamise osas ja võimaldas anda projekti kontekstis täpsemaid ajahinnanguid erinevate osade valmimise osas. Sellest tulenevalt otsustati kasutada lõputöö ülesande lahendamiseks WkHtmlToPdf rakendust.

4 PDF dokumendi genereerimise loogika arendamine

WkHtmlToPdf konsoolirakendusel baseeruva PDF dokumendi genereerimise loogika arendamine läbis lõputöö käigus kaks iteratsiooni. Esiteks kasutati rakendust läbi *Codaxy.WkHtmlToPdf* [4] teegi, mis vahendab *WkHtmlToPdf* konsoolirakendusele antavaid käsklusi C# programmeerimiskeeles. Vastav teek valiti sellepärast, et see ei lisa juurde täiendavat loogikat ega funktsionaalsust, vaid on ülesehituselt võimalikult lihtne, vähendades sellega võimalikke veatekke kohti. Ainus lisaloogika, mis selle teegiga kaasneb, on genereeritud faili sisu kirjutamine baitmassiivi või mäluvoogu, kui teegi kasutaja määrab ära konverteri väljakutsumisel *PdfOutput* objekti vastavalt kas *OutputCallback* või *OutputStream* atribuudi. Kui uue faili nime ei määrata konverteri kasutamisel, siis luuakse *WkHtmlToPdf* abil ajutine PDF dokument, mis eemaldatakse pärast konverteerimise õnnestumist.

4.1 *WkHtmlToPdf* rakenduse kasutamine *Codaxy.WkHtmlToPdf* teegi abil

Codaxy.WkHtmlToPdf teegi kasutamiseks tuli veebirakenduse konfiguratsioonifailis kirjeldada ära viide *WkHtmlToPdf* rakendusele. Iga PDF dokumendi genereerimise päringu korral käivitatakse eraldiseisev eksemplar konfiguratsioonifailis toodud viite alusel sellest programmist ja teostatakse PDF dokumendi genereerimine. Lisatud funktsionaalsuse kasutamiseks kirjutas töö autor serverirakenduses asünkroonse meetodi (Joonis 1).

Enne PDF dokumendi genereerimist veendutakse malli olemasolus. Kui genereerimise meetodisse on kaasa antud malli identifikaator, siis päritakse identifikaatori alusel PRIS-i andmebaasist viide valitud mallile. Malli identifikaatori puudumisel kontrollitakse, kas päringu tüüp on parameetrite seas kirjeldatud. Kui on, siis päritakse esimene sellele päringu tüübile vastav mall andmebaasist. Kui malli identifikaatorit ja päringu tüüpi pole meetodi parameetrites kirjeldatud, siis tagastab meetod tühiväärtuse. Vastavat kontrolli tehakse joonisel 1 esile toodud meetodis „//malli pärimise loogika ja kontroll“ kohal. *PdfConvert.ConvertHtmlToPdf* sees *PdfDocument* objekti atribuutidena määratakse Html

kohale HTML dokument, millest PDF dokumenti genereeritakse ning *FooterRight* atribuut määrab ära parempoolse joondusega dokumendi jaluse sisu. „[page]/[topage]“ määrab jaluse sisuks leheküljenumbri kuju, milles „[page]“ vastab käesoleva lehekülje numbrile ja „[topage]“ vastab kõikide lehekülgede arvule.

```
public async Task<GeneratedFile> GenerateParinguPdf(
    string paringuXml,
    long? paringuTyypId = null,
    long? mallID = null)
{
    var xmlHelper = mew ParingXmlHelper(
        this.storage,
        this.dataContextFactory);
    DokumendiMallid mall = null;

    //malli pärimise loogika ja kontroll

    var html = xmlHelper.GetParingAsHtml(
        paringuXml,
        mall.failiViitedId,
        paringuTyypId);
    var fileName = $"{mall.Nimi}_{DateTime.Today:yyMMdd}.pdf";

    byte[] pdfBytes = null;

    PdfConvert.ConvertHtmlToPdf(
        new PdfDocument {
            Html = html,
            FooterRight = "[page]/[topage]"
        },
        new PdfOutput {
            OutputCallback = delegate (
                PdfDocument doc,
                byte[] array) {
                pdfBytes = array;
            }
        }
    );
    var fileStorage = NinjectDependencyResolver
        .Resolve<FileStorage>();
    var failiViide = fileStorage.CreateFile(fileName, pdfBytes);
    return new GeneratedFile {
        FailiViitedId = failiViide,
        FileName = fileName
    };
}
```

Joonis 1. PDF dokumendi genereerimise meetod.

Codaxy.WkHtmlToPdf teegi kasutamisel joonisel 1 toodud viisil kirjutatakse genereeritud fail baitmassiivi *pdfBytes*, kasutades selleks *OutputCallback* kohal defineeritud meetodit, ja pärast seda salvestatakse baitmassiiv PDF dokumendina PRIS-i failihoidlasse ning genereeritud faili viide rakenduse andmebaasi ja meetod tagastab loodud faili viite. Kui *OutputCallback* on *PdfOutput* objektis väärtustatud, siis loeb *Codaxy.WkHtmlToPdf* teek loodud PDF dokumendi mällu baitmassiivina ja tagastab selle kasutaja poolt defineeritud *OutputCallback* meetodi päises. Genereeritud failidele antakse nimi mallile antud nimetuse ja genereerimise teostamise kuupäeva kombinatsioonina.

Loodud lahendus toimis ja probleeme ei esinenud. *WkHtmlToPdf* suutis korrektselt korrata ka tabelite päiseid lehevahetusel ning testimise käigus leitud vigade puhul seisnes probleem pigem mallide ülesehituses ja XML-ist HTML-i transformeerimisel kui *WkHtmlToPdf* i sisese loogikaga.

Klient avaldas soovi võimaluse korral vältida vajadust installeerida serveritesse eraldi rakendust, mistõttu uuriti võimalikke teeke, mis kapseldaksid *WkHtmlToPdf* rakenduse sisu DLL-faili (*Dynamic Link Library*) ja pakuksid C# programmeerimiskeeles võimalust PDF dokumendi genereerimise loogika väljakutumiseks. Esimeseks selliseks variandiks katsetati *TuesPechkin* teeki [5]. Leidus ka teisi teeke, mis võimaldaksid jooksutada PDF dokumendi genereerimist kõrvalharul, kuid enamusel neist puudus teegi kirjelduses ülevaade sellest, kas teek toimib IIS-is veebirakenduse hostimise korral, või siis oli eraldi välja toodud, et teek ei toeta seda või vastavat tuge pole testitud. Sellisteks teekideks on näiteks *DinkToPdf* [6], *WkHtmlToXSharp* [7] ja *OpenHtmlToPdf* [8].

4.2 *WkHtmlToPdf* rakenduse kasutamine *TuesPechkin* teegi abil

TuesPechkin teegi kasutamiseks eemaldati kõigepealt lisatud *Codaxy.WkHtmlToPdf* teegiga seotud failid projektist ja eemaldatud veebirakenduse konfiguratsioonifailist viide *WkHtmlToPdf* rakendusele, sest uue lahenduse korral ei ole vajalik serverisse rakendust installeerida. Seejärel asendati eelnev genereerimise loogika *PdfConvert.ConvertHtmlToPdf* osa (Joonis 1) *TuesPechkin*'i teeki kasutava koodiga (Joonis 2).

```

byte[] pdfBytes = null;
var document = new HtmlToPdfDocument
{
    GlobalSettings {
        ProduceOutline = false,
        DocumentTitle =
            $"{mall.Nimi}_{DateTime.Today:yyMMdd}",
        PaperSize = PaperKind.A4,
        Margins = {
            Unit = Unit.Millimeters,
            Left = 30,
            Right = 15,
            Top = 12,
            Bottom = 12
        }
    },
    Objects {
        new ObjectSettings {
            HtmlText = html,
            FooterSettings = new FooterSettings {
                RightText = "[page]/[topage]",
                FontSize = 10
            },
            WebSettings = {
                DefaultEncoding = "utf-8",
                LoadImages = true
            }
        }
    }
};
var tempFolderDeployment = new TempFolderDeployment();
var winAnyCPUEmbeddedDeployment = new WinAnyCPUEmbeddedDeployment(
    tempFolderDeployment);
var remotingToolset = new RemotingToolset<PdfToolset>(
    winAnyCPUEmbeddedDeployment);
IConverter pdfConverter = new ThreadSafeConverter(remotingToolset);
pdfBytes = pdfConverter.Convert(document);
remotingToolset.Unload();

```

Joonis 2. *TuesPechkin*'i teeki kasutatav PDF dokumendi genereerimise loogika.

Algselt kirjeldati *IConverter* privaatse staatilise mittemuudetava atribuudina *DocumentService* C# klassis, kus on ka joonisel 1 kirjeldatud meetod, et jälgida *TuesPechkin* teegi arendaja poolt etteantud juhiseid mitte luua uut konverterit rohkem kui üks kord veebirakenduse elutsükli jooksul. Sellega tekkis peagi probleeme hargtöötluse osas. Tekkinud probleemide lahendamiseks luuakse rakenduses siiski iga kord uus konverter. Erinevuseks on see, et pärast konverteerimist käivitatakse *TuesPechkin*'i teegis

RemotingToolset.Unload funktsioon, mis eemaldab *WkHtmlToPdf* DLL-i muutmälust, võimaldades seda ilma probleemideta uuesti kasutusele võtta.

Joonisel 2 välja toodud koodis luuakse *HtmlToPdfDocument* objekt, milles on ära kirjeldatud PDF dokumendi parameetrid. Seejärel luuakse ajutise faili hoidmiseks *TempFolderDeployment*, mis määrab ära asukoha, kuhu ajutine fail *WkHtmlToPdf* i poolt salvestatakse. Järgmiseks määratakse ära, kas käitatakse *WkHtmlToPdf* programmi 32- või 64-bitist versiooni, *winAnyCPUEmbeddedDeployment* muutujaga. Antud muutuja suudab vahet teha, kummas režiimis veebirakendus töötab, ja teeb vastavalt sellele valiku kasutatava versiooni osas. *RemotingToolset<PdfToolset>* määrab ära, kas soovitakse genereerida PDF dokumenti või näiteks pilti. Järgmiseks on võimalik valida, kas kasutatakse konverteerimisel *StandardConverter* või *ThreadSafeConverter* lähenemist. *StandardConverter*'i kasutamisel jooksutatakse konverteerimisprotsessi samas harus ja sellega on suur tõenäosus, et veebiserver võib konverteerimise ajaks hanguda. *ThreadSafeConverter* seevastu käivitab konverteerimisprotsessi eraldi harus, mis tagab veebirakenduse puhul parema töökindluse.

TuesPetchkin teegil baseeruv lahendus toimib ning testimisel vigu ei ilmnenud. Lisaks arenduspoolsele testimisele, ei tuvastatud ka kliendipoolsel testimisel loodud lahenduse osas probleeme. Diplomitöö esitamise hetkeks on PRIS-i veebirakenduses selles peatükis kirjeldatud lahendus kasutusel lõppkasutajatele avaldatud versioonis.

5 X-tee päringu tulemusest HTML dokumendi loomine

X-tee päringu tulemustest HTML dokumendi loomiseks loodi eraldi C# klass nimega `ParingXmlHelper.cs`, mis kasutab ära X-tee adapteris päringute tegemiseks eksisteerivaid C# klasside XML serialiseerijaid ja deserialiseerijaid. Päringu tulemuse saabumisel PRIS-i serveripoolsesse rakendusse serialiseeritakse saabunud vastus uuesti XML-i ja see salvestatakse teksti kujul PRIS-i serveris kohaliku andmeedastusobjekti atribuudi `ParingXML` kohale. Mainitud serialiseerimist teostab selleks ettenähtud meetod (Joonis 3).

```
public static string GetResponseAsXml(
    object response,
    string paringuId
)
{
    using (var ms = new MemoryStream())
    {
        new XteeSerializer().Serialize(ms, response);
        using (var xmlstream = new MemoryStream()) {
            ms.Rewind();
            var doc = XDocument.Load(ms);
            var root = doc.Root;
            root.Add(new XElement("paringuid", paringuId));
            doc.Save(xmlstream);
            xmlstream.Rewind();
            return xmlstream.ToUtf8String();
        }
    }
}
```

Joonis 3. *GetResponseAsXml* meetod päringu tulemuse XML-i serialiseerimiseks.

Serialiseerimisel loetakse saadud XML uude mäluvoogu. Sellele järgnevalt keritakse mäluvoog algusesse ja XML loetakse *XDocument* objektina „doc“ muutujasse, võimaldades selle abil lisada XML-i juurelemendile uus alamelement, mis sisaldab unikaalset X-tee päringu identifikaatorit. Pärast identifikaatori lisamist salvestatakse muudetud XML uude mäluvoogu „xmlstream“. Selleks et tagastada loodud XML tekstiväärtusena, keritakse pärast salvestamist „xmlstream“ mäluvoog tagasi algusesse ja tagastatakse mäluvoo sisu UTF-8 vorminguga tekstina. Põhjus, miks vastavat

serialiseerimist tehakse iga kord, kui kasutaja X-tee päringu teostab, ja mitte ainult siis, kui PDF-i genereerimise päring tehakse, on see, et selle tulemusena välditakse liigseid teisendamisi PRIS-i veebirakenduse sisesteks andmeedastusobjektideks ja teistpidi X-tee teenustest saadavateks objektideks. Kui X-tee päring tagastab vea, siis tagastatakse XML-ina veateade koos päringu identifikaatoriga joonisel 4 kirjeldatud meetodi abil.

```
public static string GetEmptyResponseXml(
    string paringuId,
    string faultString
)
{
    var doc = new XDocument(
        new XElement(
            "keha",
            new XElement("paringuid", paringuId),
            new XElement("faultString", faultString)
        )
    );
    return doc.ToString();
}
```

Joonis 4. *GetEmptyResponseAsXml* meetod XML-i tagastamiseks, kui X-tee päring tagastab vea.

Kui kasutaja soovib genereerida PDF-formaadis dokumenti, siis joonisel 1 väljatoodud meetodis kutsutakse XML-ist HTML dokumendi saamiseks välja *GetParingAsHtml* meetodit (Joonis 5), millele antakse parameetritena kaasa eelnevalt loodud XML teksti kujul, viide XSL mallile ja päringu tüübi identifikaator. Enne XSL malli rakendamist lisatakse *AddHeader* meetodiga päringu XML-i juurelemendile alamelemendiks „header“ element, milles kirjeldatakse alamelementidena rakenduse keskkonna nimetus, kui PRIS-i rakendus jookseb testimis- või arenduskeskkonnas, teostatud päringu nimetus, mille väärtus päritakse „paringuTyypId“ parameetri abil andmebaasist, teostaja asutuse nimetus ja päringu teostamise kuupäev ning luuakse uus *XmlDocument* tüüpi muutuja „xml“.

Järgmise sammuna loetakse modifitseeritud XML mäluvoogu „paringStream“ ja loetakse malli viite abil XSL fail „xslFileStream“ failivoogu. Seejärel initsialiseeritakse XSL transformeerija „xslt“ XSL failiga ja keritakse „paringStream“ mäluvoog algusesse ning luuakse ja initsialiseeritakse *StringBuilder* objekt. Kasutades *XmlReader* ja *XmlWriter* objekte, loetakse „paringStream“ sisu „xmlReader“ lugejasse ja määratakse XML-i kirjutaja „xmlWriter“ väljundi asukohaks eelnevalt loodud „stringBuilder“ objekt.

Seejärel teostatakse XML-i transformeerimine HTML dokumendiks XSL transformeerija kaudu. Pärast transformatsiooni kirjutatakse loodud HTML „xml“ muutujasse.

```
public string GetParingAsHtml(
    string paringXml,
    int? xslViide,
    long? paringuTyypId)
{
    var xml = new XmlDocument();
    paringXml = this.AddHeader(paringXml, paringuTyypId);
    using (var paringStream =
        new MemoryStream(Encoding.UTF8.GetBytes(paringXml)))
    {
        using (var xslFileStream =
            this.xslStorage.LoadStream(xslViide))
        {
            var xslFile = new XmlTextReader(xslFileStream);
            xslFile.Read();
            var xslt = new XslCompiledTransform();
            xslt.Load(xslFile);
            var stringBuilder = new StringBuilder();
            paringStream.Rewind();
            using (var xmlReader =
                XmlReader.Create(paringStream))
            {
                using (var xmlWriter =
                    XmlWriter.Create(stringBuilder))
                {
                    xslt.Transform(xmlReader, xmlWriter);
                }
            }
            xml.LoadXml(stringBuilder.ToString());
        }
    }
    foreach (XmlNode node in xml)
    {
        if (node.NodeType == XmlNodeType.XmlDeclaration)
        {
            xml.RemoveChild(node);
        }
    }
    return RemoveNamespaces(xml.OuterXml);
}
```

Joonis 5. *GetParingAsHtml* meetod XML-i HTML-i transformeerimiseks.

Kui HTML on loodud, siis järgnevalt eemaldatakse sellest kõik *XmlDeclaration* (XML-i deklaratsioon) tüüpi elemendid ja täiendavalt eemaldatakse *RemoveNameSpaces* funktsiooniga HTML-ist kõik XML-i nimeruumi atribuudid. Antud toiming on oluline

sellepärast, et kui nimeruumi atribuute ei eemaldata, siis see takistab hiljem PDF-formaadis dokumendi genereerimisel CSS-i (*Cascading Style Sheet*) korrektset rakendumist.

6 X-tee päringutele XSL mallide loomine

XSL mallide loomisel tuli arvestada iga X-tee päringu tulemusel korral XML-i sisust tulenevate erinevustega ja kliendi poolt esitatud nõuetega lõplike dokumentide vorminduse osas. Suures osas rakendus igale mallile sarnane laadistik (Joonis 6), kuid mitmete puhul tuli lisada nõutud kujundusest tulenevaid laadi kirjeldusi. Nõuete kohaselt on kogu HTML sisule üldine tähemärgi suurus 12 pikslit ja teksti fondina *Arial* ja tabelite ridade keskel ei tohi olla lehekülje vahetust, nagu on kirjeldatud „tr“ stiiliosas.

```
body {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 12px;
}
tr {
    page-break-inside: avoid;
}
```

Joonis 6. XSL mallide üldine kaskaadlaadistik.

Lisaks üldisele laadistikule esines probleeme kuupäevade kuvamisega soovitud vormingus. Probleemi lahendamiseks lisas töö autor XSL mallidesse viite *Microsoft*'i poolt arendatud stiililehele, mis sisaldab kuupäevade vormindamise funktsionaalsust XML-i transformeerimisel XSL malli abil. Mõningaid probleeme esines ka algselt tingimuslausete ülesehitusega, kuid need olid lahendatavad *Mozilla Developer Network*'i veebilehel XSLT transformeerija dokumentatsiooni [10] abil. Lõputöö lisades 1, 2 ja 3 on toodud näited lõplikest mallidest, mis loodi arenduse käigus vastavalt äriregistri ettevõtja lihtandmete ja liiklusregistrist kehtivate juhilubade ja sõiduki detailandmete päringule.

Kokku loodi arenduse käigus 29 erinevat malli. Iga malli sisuosa kujundus on unikaalne ja vastab kliendi poolt esitatud päringupõhiste nõuetele. Lõputöö lisades 4 ja 5 on vastavalt välja toodud arenduskeskkonnas testandmetega tehtud kinnistu detailandmete ning sõiduki detailandmete päringu põhjal genereeritud PDF dokument.

7 Kokkuvõte

Lõputöö põhieesmärk oli PRIS-i näitel luua lahendus X-tee päringu tulemustest PDF-formaadis dokumentide loomiseks.

Töö teoreetilises osas võrreldi erinevaid lähenemisviise ja kasutatavaid teeke. Tulenevalt ülesande lähtetingimustest otsustas töö autor arendada PDF-formaadis dokumentide genereerimise loogika PRIS-i veebirakenduse serveripoolsesse rakendusse, kasutades selleks *TuesPechkin* teeki, mis kasutab PDF dokumendi loomiseks HTML dokumendist *WkHtmlToPdf* konsoolirakendust. Antud teek võimaldab täita kliendi poolt genereeritavatele dokumenditele esitatud vormindusnõudeid ja tagab rakenduse töökindluse. X-tee päringu tulemustest HTML-i loomiseks kasutas töö autor igale päringu tüübile vastavaid malle. HTML dokumendi loomisel tuleb arvestada, et *WkHtmlToPdf* rakendus ei suuda tagada kaskaadlaadistiku korrektset rakendumist, kui HTML ei ole eelnevalt puhastatud, eemaldades üleliigsed XML-i deklaratsiooni elemendid ja XML nimeruumi atribuudid elementidelt.

Töö tulemuseks on töötav PDF-formaadis dokumentide genereerimise loogika, mis on kasutusel PRIS-i lõppkasutajatele ligipääsetavas versioonis ja 29 unikaalset XSL malli, millest iga mall vastab ühele X-tee päringu tüübile ja arvestab kliendi poolt esitatud kujunduse nõudeid vastavast päringu tulemusest loodavale PDF dokumendile.

Kasutatud kirjandus

- [1] Riigi Infosüsteemide Amet, „X-tee rakendusjuhhis,“ Riigi Infosüsteemide Amet, 16 08 2016. [Võrgumaterjal]. Available: <https://moodle.ria.ee/mod/page/view.php?id=180>. [Kasutatud 27 04 2020].
- [2] A. Teplitzki ja J. M. M. Poo, „HtmlRenderer,“ [Võrgumaterjal]. Available: <https://archive.codeplex.com/?p=htmlrenderer>. [Kasutatud 19 04 2020].
- [3] J. Schou-Rode, „iTextSharp (LGPL / MPL) 4.1.6,“ 8 12 2014. [Võrgumaterjal]. Available: <https://github.com/schourode/iTextSharp-LGPL>. [Kasutatud 26 04 2020].
- [4] J. Truelsen ja A. Kulkarni, „wkhtmltopdf,“ [Võrgumaterjal]. Available: <https://wkhtmltopdf.org/>. [Kasutatud 19 04 2020].
- [5] M. Stijak, „Codaxy.WkHtmlToPdf,“ [Võrgumaterjal]. Available: <https://github.com/codaxy/wkhtmltopdf>. [Kasutatud 26 04 2020].
- [6] D. Gray, „TuesPechkin,“ [Võrgumaterjal]. Available: <https://github.com/tuespetre/TuesPechkin>. [Kasutatud 26 04 2020].
- [7] R. Dvojmoč, „DinkToPdf,“ [Võrgumaterjal]. Available: <https://github.com/rdvojmoc/DinkToPdf>. [Kasutatud 26 04 2020].
- [8] P. R. García, „WkHtmlToXSharp,“ [Võrgumaterjal]. Available: <https://github.com/pruiz/WkHtmlToXSharp>. [Kasutatud 26 04 2020].
- [9] T. Vilppu, „OpenHtmlToPdf,“ [Võrgumaterjal]. Available: <https://github.com/vilppu/OpenHtmlToPdf>. [Kasutatud 26 04 2020].
- [10] Mozilla Developer Network Contributors, „Transforming XML with XSLT,“ Mozilla, 18 03 2019. [Võrgumaterjal]. Available: https://developer.mozilla.org/en-US/docs/Web/XSLT/Transforming_XML_with_XSLT. [Kasutatud 26 04 2020].

Lisa 1 – Ettevõtja lihtandmete päringu XSL mall

```
<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="2.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:ms="urn:schemas-
microsoft-com:xslt">
  <xsl:template match="/">
    <html>
      <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8"/>
        <style>
          body {
            font-family: Arial, Helvetica, sans-
serif;
            font-size: 12px;
          }
          table, th, td {
            border: 1px solid black;
            border-collapse: collapse;
            font-size: 12px;
          }
          th {
            text-align: left;
          }
          .paringu-id {
            padding-top:10px;
          }
          thead {
            display: table-header-group;
          }
          tfoot {
            display: table-row-group;
          }
          tr {
            page-break-inside: avoid;
          }
        </style>
      </head>
      <body>
        <xsl:apply-templates select="keha" mode="root"/>
      </body>
    </html>
  </xsl:template>
<xsl:template match="keha" mode="root">
```

```

        <xsl:apply-templates select="header"/>
        <div>
            <div class="paringu-id">
                <small>Päringu ID: <xsl:value-of
select="paringuid"/></small>
            </div>
            <xsl:apply-templates select="paring"/>
            <h2 style="border-bottom:1px solid;width:100%;">Päringu
vastus</h2>
            <div>
                <xsl:apply-templates select="keha"
mode="response"/>
            </div>
        </div>
    </xsl:template>
    <xsl:template match="keha" mode="response">
        <xsl:if test="not(ettevotjad/item)">
            <div>Andmeid ei leitud</div>
        </xsl:if>
        <xsl:if test="ettevotjad/item">
            <p><b>Leitud ettevõtete arv: </b><xsl:value-of
select="leitud_ettevotjate_arv"/></p>
            <xsl:apply-templates select="ettevotjad"/>
        </xsl:if>
    </xsl:template>
    <xsl:template match="ettevotjad">
        <xsl:for-each select="item">
            <div>
                <table class="table" width="100%">
                    <tr>
                        <td><b>Ettevõtte nimi</b></td>
                        <td><xsl:value-of
select="evnimi"/></td>
                        <td><b>Esmakande aeg</b></td>
                        <td><xsl:value-of
select="esmakande_aeg"/></td>
                    </tr>
                    <tr>
                        <td><b>Õiguslik vorm</b></td>
                        <td><xsl:value-of
select="oiguslik_vorm_tekstina"/></td>
                        <td><b>Ettevõtte staatus</b></td>
                        <td><xsl:value-of
select="staatus_tekstina"/></td>
                    </tr>
                    <tr>
                        <td><b>Registrikood</b></td>
                        <td><xsl:value-of
select="ariregistri_kood"/></td>
                        <td><b>Registri kustutamise
aeg</b></td>

```

```
 <xsl:value-of select="registrist_kustutamise_aeg"/></td> </tr> <tr>  <b>Piirkond</b></td>  <xsl:value-of select="piirkond_tekstina"/></td>  <b>Kapitali suurus</b></td>  <xsl:value-of select="concat(evkapitalid/kapitali_suurus, ' ',evkapitalid/kapitali_valuuta)"/></td> </tr> <tr>  <b>Ettevõtte aadress</b></td>  <xsl:value-of select="evaadressid/asukoht_ettevotja_aadressis"/></td>  <b>Asutatud sissemakset tegemata</b></td>  <xsl:variable name="keel" select="../../../paring/keel"/> <xsl:if test="evkapitalid/asutatud_sissemakset_tegemata = 'true'"> <xsl:if test="$keel = 'eng'">Yes</xsl:if> <xsl:if test="$keel != 'eng'">Jah</xsl:if> </xsl:if> <xsl:if test="evkapitalid/asutatud_sissemakset_tegemata = 'false'"> <xsl:if test="$keel = 'eng'">No</xsl:if> <xsl:if test="$keel != 'eng'">Ei</xsl:if> </xsl:if> </td> </tr> <tr>  <b>EHAK</b></td>  <xsl:if test="evaadressid/asukoha_ahak_kood/text()"> <xsl:value-of select="concat(evaadressid/asukoha_ahak_kood, ' / ',evaadressid/asukoha_ahak_tekstina)"/> </xsl:if> </td>  <b>Postiindeks</b></td>  <xsl:value-of select="evaadressid/indeks_ettevotja_aadressis"/></td> | | | | | | | | | | | | |
```

```

        </tr>
    </table>
    <xsl:apply-templates select="kehtetud_nimed"/>
</div>
<br/>
</xsl:for-each>
</xsl:template>
<xsl:template match="kehtetud_nimed">
    <xsl:if test="evnimi">
        <table class="table" width="100%" border="1"
style="border-collapse:collapse;">
            <thead>
                <tr>
                    <th style="text-align:left;"><b>Kehtetud ärinimed</b></th>
                    <th style="text-align:left;"><b>Alguskuupäev</b></th>
                    <th style="text-align:left;"><b>Lõppkuupäev</b></th>
                </tr>
            </thead>
            <tbody>
                <xsl:for-each select="evnimi">
                    <tr>
                        <td><xsl:value-of
select="evnimi"/></td>
                        <td><xsl:value-of
select="ms:format-date(algus_kpv,'dd.MM.yyyy')"/></td>
                        <td><xsl:value-of
select="ms:format-date(lopp_kpv,'dd.MM.yyyy')"/></td>
                    </tr>
                </xsl:for-each>
            </tbody>
        </table>
    </xsl:if>
</xsl:template>
<xsl:template match="header">
    <div class="paring-header">
        <h1 style="color:red;text-transform:uppercase;"><xsl:value-of select="keskkond"/></h1>
        <h1>Äriregister</h1>
        <div><small>Registrite ja Infosüsteemide Keskus, Lubja
4,19081 Tallinn</small></div>
        <br/>
        <div><xsl:value-of select="concat('Päringu nimetus:
',paringu_nimetus)"/></div>
        <div><xsl:value-of select="concat('Päringu kuupäev:
',kuupaev)"/></div>
        <div><xsl:value-of select="concat('Päringu teostaja:
',teostaja)"/></div>
    </div>
</xsl:template>

```

```

    <xsl:template match="paring">
      <div>
        <h2 style="border-bottom:1px solid;width:100%;">Päringu
sisend</h2>
        <xsl:for-each select="node()">
          <xsl:if test="* or (text() and text() != 0)">
            <xsl:for-each select="node()">
              <xsl:if test="text()">
                <div>
                  <span><xsl:value-of
select="name(parent:*)" />.<xsl:value-of select="name()" />: <xsl:value-of
select="text()" /></span>
                  </div>
                </xsl:if>
              </xsl:for-each>
            </xsl:if>
          <xsl:if test="text()">
            <div>
              <span><xsl:value-of select="name()" />:
<xsl:value-of select="text()" /></span>
            </div>
          </xsl:if>
        </xsl:for-each>
      </div>
    </xsl:template>
  </xsl:stylesheet>

```

Lisa 2 – Kehtivate juhulubade päringu XSL mall

```
<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="2.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:ms="urn:schemas-
microsoft-com:xslt">
  <xsl:template match="/">
    <html>
      <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8"/>
        <style>
          body {
            font-family: Arial, Helvetica, sans-
serif;
            font-size: 12px;
          }
          .border, .border th, td {
            border: 1px solid black;
            border-collapse: collapse;
            font-size: 12px;
          }
          th {
            text-align: left;
          }
          .paringu-id {
            padding-top:10px;
          }
          .no-border {
            border-style:hidden;
          }
          .no-border td, th {
            border-style:hidden;
            font-size: 12px;
          }
          <thead {
            display: table-header-group;
          }
          <tfoot {
            display: table-row-group;
          }
          <tr {
            page-break-inside: avoid;
          }
        </style>
      </head>
```

```

        <body>
            <xsl:apply-templates select="keha"/>
        </body>
    </html>
</xsl:template>
<xsl:template match="keha">
    <xsl:apply-templates select="header" />
    <div>
        <div class="paringu-id">
            <small>Päringu ID: <xsl:value-of
select="paringuid"/></small>
        </div>
        <xsl:apply-templates select="paring"/>
        <h2 style="border-bottom:1px solid;width:100%;">Päringu
vastus</h2>
        <xsl:apply-templates select="juhid" />
    </div>
</xsl:template>
<xsl:template match="juhid">
    <xsl:if test="not(item)">
        <div>Andmeid ei leitud</div>
    </xsl:if>
    <xsl:if test="item">
        <table class="no-border">
            <xsl:for-each select="item">
                <tr>
                    <td><strong>Isikukood:</strong></td>
                    <td><xsl:value-of
select="isikukood"/></td>
                </tr>
                <tr>
                    <td><strong>Perekonnanimi:</strong></td>
                    <td><xsl:value-of
select="perekonnanimi"/></td>
                </tr>
                <tr>
                    <td><strong>Eesnimi:</strong></td>
                    <td><xsl:value-of
select="eesnimi"/></td>
                </tr>
                <tr>
                    <td><strong>Sünniaeg:</strong></td>
                    <td><xsl:value-of
select="synniaeg"/></td>
                </tr>
                <tr>
                    <td><strong>Aadress:</strong></td>
                    <td><xsl:value-of
select="aadress"/></td>
                </tr>
            </xsl:for-each>
        </table>
    </xsl:if>
</xsl:template>

```

```

riik:</strong></td>
select="synnikoha_riik"/></td>
</tr>
<tr>
<td><strong>Sünnikoha
<td><xsl:value-of
select="synnikoha_riik"/></td>
</tr>
<tr>
<td><strong>Haldusüksus:</strong></td>
<td><xsl:value-of
select="haldusyksus"/></td>
</tr>
<tr>
<xsl:apply-templates
select="juhiloa"></xsl:apply-templates>
</tr>
</xsl:for-each>
</table>
</xsl:if>
</xsl:template>
<xsl:template match="juhiloa">
<table class="border" width="100%">
<thead>
<tr>
<th>Juhiloa nr</th>
<th>Loa tüüp</th>
<th>Kategooriad</th>
<th>Välja antud</th>
<th>Kätte antud</th>
<th>Kehtivuse lõpp</th>
<th>Staatus</th>
</tr>
</thead>
<tbody>
<xsl:for-each select="item">
<tr>
<td><xsl:value-of
select="juhiloanr"/></td>
<td><xsl:value-of
select="loatyyp"/></td>
<td><xsl:value-of
select="kategooriad"/></td>
<td><xsl:value-of
select="valjaantud"/></td>
<td><xsl:value-of
select="katteantud"/></td>
<td><xsl:value-of
select="kehtiv_lopp"/></td>
<td><xsl:value-of
select="staatus"/></td>
</tr>
</xsl:for-each>
</tbody>
</table>

```

```

</xsl:template>
<xsl:template match="header">
  <div class="paring-header">
    <h1 style="color:red;text-
transform:uppercase;"><xsl:value-of select="keskkond"/></h1>
    <h1>Liiklusregister</h1>
    <p><small>Maanteeamet, Teelise 4,10916
Tallinn</small></p>
    <br/>
    <div><xsl:value-of select="concat('Päringu nimetus:
',paringu_nimetus)"/></div>
    <div><xsl:value-of select="concat('Päringu kuupäev:
',kuupaev)"/></div>
    <div><xsl:value-of select="concat('Päringu teostaja:
',teostaja)"/></div>
  </div>
</xsl:template>
<xsl:template match="paring">
  <div>
    <h2 style="border-bottom:1px solid;width:100%;">Päringu
sisend</h2>
    <xsl:for-each select="node()">
      <xsl:if test="* or (text() and text() != 0)">
        <xsl:for-each select="node()">
          <xsl:if test="text()">
            <div>
              <span><xsl:value-of
select="name(parent::*)">.<xsl:value-of select="name()"/>: <xsl:value-of
select="text()"/></span>
            </div>
          </xsl:if>
        </xsl:for-each>
      </xsl:if>
    <xsl:if test="text()">
      <div>
        <span><xsl:value-of select="name()"/>:
<xsl:value-of select="text()"/></span>
      </div>
    </xsl:if>
  </xsl:for-each>
</div>
</xsl:template>
</xsl:stylesheet>

```

Lisa 3 – Sõiduki detailandmete päringu XSL mall

```
<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="2.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:ms="urn:schemas-
microsoft-com:xslt">
  <xsl:template match="/">
    <html>
      <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8"/>
        <style>
          body {
            font-family: Arial, Helvetica, sans-
serif;
            font-size: 12px;
          }
          th {
            text-align: left;
            font-size: 12px;
          }
          .paringu-id {
            padding-top:10px;
          }
          .no-border {
            border-style:hidden;
          }
          .no-border td, th {
            border-style:hidden;
            font-size: 12px;
          }
          thead {
            display: table-header-group;
          }
          tfoot {
            display: table-row-group;
          }
          tr {
            page-break-inside: avoid;
          }
          hr {
            border:none;
            height: 1px;
            border-bottom:solid 1px;
          }
        </style>
      </head>
    </html>
  </template>
</xsl:stylesheet>
```

```

        </style>
    </head>
    <body>
        <xsl:apply-templates select="keha"/>
    </body>
</html>
</xsl:template>
<xsl:template match="keha">
    <xsl:apply-templates select="header" />
    <div>
        <div class="paringu-id">
            <small>Päringu ID: <xsl:value-of
select="paringuid"/></small>
        </div>
        <xsl:apply-templates select="paring"/>
        <h2 style="border-bottom:1px solid;width:100%;">Päringu
vastus</h2>
        <xsl:apply-templates select="paring2Response" />
    </div>
</xsl:template>
<xsl:template match="paring2Response">
    <xsl:if test="not(item)">
        <div>Andmeid ei leitud</div>
    </xsl:if>
    <xsl:if test="item">
        <xsl:for-each select="item">
            <xsl:if test="position() > 1">
                <hr></hr>
            </xsl:if>
            <table class="no-border">
                <tr>
                    <td><strong>Registreerimismärk:</strong></td>
                    <td><xsl:value-of
select="regmrk"/></td>
                    <td><strong>Registreerimistunnistuse
number:</strong></td>
                    <td><xsl:value-of
select="regtun"/></td>
                </tr>
                <tr>
                    <td><strong>Registreerimismärgi
tüüp:</strong></td>
                    <td><xsl:value-of
select="nr_tyyp"/></td>
                    <td><strong>Registreerimistunnistuse
väljastamise kuupäev:</strong></td>
                    <td><xsl:value-of
select="regtun_kuup"/></td>
                </tr>
                <tr>
                    <td><strong>ID-kood:</strong></td>

```

```

select="idkood"/></td>
kuupäev:</strong></td>
select="regtulek"/></td>
</tr>
<tr>
(tehasetähis):</strong></td>
select="vinkood"/></td>
väljastaja andmed:</strong></td>
select="regtun_valjastaja"/></td>
</tr>
<tr>
style="float:left;width:50%;">
select="soiduki_andmed"/>
style="float:right;width:50%;">
select="oma_andmed_v"/>
<table>
<tr>
<td><strong>Sõiduki kasutaja</strong></td>
</tr>
<tr>
<td><xsl:value-of select="kasutajad"/></td>
</tr>
</table>
</div>
</td>
</tr>
</table>
</xsl:for-each>
</xsl:if>
</xsl:template>
<xsl:template match="soiduki_andmed">
<table class="no-border">
<tr>
<td colspan="2"><strong>Sõiduki
andmed</strong></td>
</tr>
<tr>
<td><strong>Kategooria:</strong></td>

```

```

        <td><xsl:value-of select="kateg"/></td>
    </tr>
    <tr>
        <td><strong>Mark:</strong></td>
        <td><xsl:value-of select="marknm"/></td>
    </tr>
    <tr>
        <td><strong>Mudel:</strong></td>
        <td><xsl:value-of select="mudel_mod"/></td>
    </tr>
    <tr>
        <td><strong>Kere tüüp:</strong></td>
        <td><xsl:value-of select="kerenm"/></td>
    </tr>
    <tr>
        <td><strong>Mootori tüüp:</strong></td>
        <td><xsl:value-of select="mootornm"/></td>
    </tr>
    <tr>
        <td><strong>Mootori maht:</strong></td>
        <td><xsl:value-of select="mootmht"/></td>
    </tr>
    <tr>
        <td><strong>Mootori võimsus:</strong></td>
        <td><xsl:value-of select="mootvms"/></td>
    </tr>
    <tr>
        <td><strong>Ehitusaasta:</strong></td>
        <td><xsl:value-of select="ehitaasta"/></td>
    </tr>
    <tr>
        <td><strong>Värvus:</strong></td>
        <td><xsl:value-of select="varv_nimetus"/></td>
    </tr>
</table>
</xsl:template>
<xsl:template match="oma_andmed_v">
    <table class="no-border">
        <tr>
            <td colspan="2"><strong>Sõiduki
omanik</strong></td>
        </tr>
        <tr>
            <td><strong>Isiku- või äriregistri
kood:</strong></td>
            <td><xsl:value-of select="oma_kood"/></td>
        </tr>
        <tr>
            <td><strong>Nimi või perekonnanimi:</strong></td>
            <td><xsl:value-of select="concat(oma_nimi, ' ',
oma_eesnimi)"/></td>
        </tr>
    </table>

```

```

        </tr>
        <tr>
            <td><strong>Elu- või asukoht:</strong></td>
            <td><xsl:value-of select="aadress"/></td>
        </tr>
        <tr>
            <td><strong>Asukoha kood:</strong></td>
            <td><xsl:value-of select="asuk_akood"/></td>
        </tr>
        <tr>
            <td><strong>Muud andmed:</strong></td>
            <td><xsl:value-of select="oma_muud"/></td>
        </tr>
    </table>
</xsl:template>
<xsl:template match="header">
    <div class="paring-header">
        <h1 style="color:red;text-
transform:uppercase;"><xsl:value-of select="keskkond"/></h1>
        <h1>Liiklusregister</h1>
        <p><small>Maanteeamet, Teelise 4,10916
Tallinn</small></p>
        <br/>
        <div><xsl:value-of select="concat('Päringu nimetus:
',paringu_nimetus)"/></div>
        <div><xsl:value-of select="concat('Päringu kuupäev:
',kuupaev)"/></div>
        <div><xsl:value-of select="concat('Päringu teostaja:
',teostaja)"/></div>
    </div>
</xsl:template>
<xsl:template match="paring">
    <div>
        <h2 style="border-bottom:1px solid;width:100%;">Päringu
sisend</h2>
        <xsl:for-each select="node()">
            <xsl:if test="* or (text() and text() != 0)">
                <xsl:for-each select="node()">
                    <xsl:if test="text()">
                        <div>
                            <span><xsl:value-of
select="name(parent::*)"/>.<xsl:value-of select="name()"/>: <xsl:value-of
select="text()"/></span>
                        </div>
                    </xsl:if>
                </xsl:for-each>
            </xsl:if>
            <xsl:if test="text()">
                <div>
                    <span><xsl:value-of select="name()"/>:
<xsl:value-of select="text()"/></span>
                </div>
            </xsl:if>
        </xsl:for-each>
    </div>

```

```
                </xsl:if>
            </xsl:for-each>
        </div>
    </xsl:template>
</xsl:stylesheet>
```

Lisa 4 – Arenduskeskkonnas testandmetega loodud kinnistu detailandmete päringule vastav PDF dokument

RIK ARENDUSED

Kinnistusraamat

Juditsministeerium, Suur-Ameerika 1, 10122 Tallinn

Päringu nimetus: Kinnistu detailandmete päring

Päringu kuupäev: 28.02.2020

Päringu teostaja: Põhja Ringkonnaprokuratuur

Päringu ID: QWTJiWNNZrLJ0xXelv+nGAG6BvGySNSZK3fM=

Päringu vastus

Üldandmed

| | |
|----------------------------|--------------------------------------|
| Kinnistu number | 496138 |
| Kinnistuosakond | Põhja, Tartu Maakohu kinnistuosakond |
| Korteri nr | |
| Korteriühistu registrikood | |
| Digitõimik | Jah |

I jagu - KINNISTU KOOSSES

| Kande number | Katastritunnus | Sihtotstarve ja asukoht | Pindala Info | Kande kehtivus | |
|--------------|----------------|---|--------------|---|------------|
| 1 | 85602:002:0611 | maatulundusmaa, Valgärne vald, Pikareinu küla. | 4,5 ha | Registriosa avamisel 29.01.1998 kinnistamisavalduse alusel sisse kantud 24.03.1998; Omber kirjutatud 22.02.2002. Kohtunikuabi A. Luur | kustutatud |
| 1 | 85602:002:0611 | Maatulundusmaa 100%, Põlvamaa, Valgärne vald, Pikareinu küla, Peelijä. | 4,5 ha | Maakatastri andmed üle võetud 13.06.2010. | kustutatud |
| 1 | 85602:002:0169 | Maatulundusmaa 100%, Põlva maakond, Valgärne vald, Pikareinu küla, Peelijä. | 16,08 ha | Maakatastri andmed üle võetud 29.08.2012. | kustutatud |
| 1 | 85602:002:0169 | Maatulundusmaa 100%, Põlva maakond, Valgärne vald, Pikareinu küla, Peelijä. | 16,08 ha | | kustutatud |
| 1 | 85602:002:0169 | Maatulundusmaa 100%, Põlva maakond, Valgärne vald, Pikareinu küla, Peelijä. | 16,08 ha | Maakatastri andmed üle võetud 27.09.2017. | kustutatud |
| 1 | 85602:002:0169 | Maatulundusmaa 100%, Põlva maakond, Kanepi vald, Pikareinu küla, Peelijä. | 16,08 ha | Maakatastri andmed üle võetud 15.12.2017. | kustutatud |
| 1 | 85602:002:0169 | Maatulundusmaa 100%, Põlva maakond, Kanepi vald, Pikareinu küla, Peelijä. | 16,08 ha | Maakatastri andmed üle võetud 24.01.2018. | kustutatud |
| 1 | 85602:002:0169 | Maatulundusmaa 100%, Põlva maakond, Kanepi vald, Pikareinu küla, Peelijä. | 16,08 ha | Maakatastri andmed üle võetud 23.12.2018. | kustutatud |
| 1 | 85602:002:0169 | Maatulundusmaa 100%, Põlva maakond, Kanepi vald, Pikareinu küla, Peelijä. | 16,08 ha | Maakatastri andmed üle võetud 31.12.2018. | kehtiv |

II jagu - OMANIK

| Kande number | Omanik | Kande alus | Kande kehtivus |
|--------------|--|------------|----------------|
| 1 | Voldemar Agro (32808166514, EST) 1 - Anuõigustatus | | kustutatud |
| 2 | Heino Agro (35710266524, EST) 1 - Anuõigustatus | | kehtiv |

III jagu - KOORMATISED JA KITSENDUSED

IV jagu - HÜPOTEEGID

Lisa 5 – Arenduskeskkonnas testandmetega loodud sõiduki detailandmete päringule vastava PDF dokumendi esimesed 2 lehekülge

| RIK ARENDUSED | | | |
|---|------------------------------------|---|---|
| Liiklusregister | | | |
| Maanteeamet, Teelise 4, 10916 Tallinn | | | |
| Päringu nimetus: Sõiduki detailandmete päring | | | |
| Päringu kuupäev: 09.03.2020 | | | |
| Päringu loostaja: Põhja Ringkonnaprokuratuur | | | |
| Päringu ID: qe9dH3RTx08f+U64dGEO3agjDG2MDa2ba0ashz0Y= | | | |
| Päringu vastus | | | |
| Registreerimismärk: | 102BLM | Registreerimistunnistuse number: | EL594865 |
| Registreerimismärgi tüüp: | A1 – Üldkasutatav auto | Registreerimistunnistuse väljastamise kuupäev: | 04.02.2019 |
| ID-kood: | A9181228009 | Esmase registreerimise kuupäev: | 04.02.2019 |
| VIN-kood (tehasetähis): | MVCJYKL10JH021295 | Registreerimistunnistuse väljastaja andmed: | Eesti Riiklik Autoregistrikeskus |
| Sõiduki andmed | | Sõiduki omanik | |
| Kategooria: | N1G | Isiku- või äriregistri kood: | 47003280318 |
| Mark: | MTSUBISHI | Nimi või perekonnanimi: | MAANTEE-ÕIS-LEHEMETS JULIA |
| Mudel: | L200 | Elu- või asukoht: | Harju maakond, Lääne-Harju vald, Tuulna küla, Põhja-Pauli tee 3 |
| Kere tüüp: | PIKAP | Asukoha kood: | 8460 |
| Mootori tüüp: | DIISEL | Muud andmed: | KEHTIV |
| Mootori maht: | 2442 | Sõiduki kasutaja | |
| Mootori võimsus: | 133 | | |
| Ehitusaasta: | 2019 | | |
| Värvus: | ROHELINE | | |
| Registreerimismärk: | 12HPG | Registreerimistunnistuse number: | EL594868 |
| Registreerimismärgi tüüp: | A3H - Vahendatud mõõlmetega haagis | Registreerimistunnistuse väljastamise kuupäev: | 08.08.2019 |
| ID-kood: | MB190808002 | Esmase registreerimise kuupäev: | 08.08.2019 |
| VIN-kood (tehasetähis): | ARKEST20190808001 | Registreerimistunnistuse väljastaja andmed: | Eesti Riiklik Autoregistrikeskus |
| Sõiduki andmed | | Sõiduki omanik | |
| Kategooria: | O1 | Isiku- või äriregistri kood: | 47003280318 |
| Mark: | OMVA/ALMSTATUD | Nimi või perekonnanimi: | MAANTEE-ÕIS-LEHEMETS JULIA |
| Mudel: | OMVA/ALMSTATUD | Elu- või asukoht: | Saare maakond, Saaremaa vald, Aru küla, Põldmaa |
| Kere tüüp: | PANDIVECK | Asukoha kood: | 1364 |
| Mootori tüüp: | | Muud andmed: | KEHTIV |
| Mootori maht: | | Sõiduki kasutaja | |
| Mootori võimsus: | | | |
| Ehitusaasta: | 2019 | | |
| Värvus: | ORANZ | | |
| Registreerimismärk: | 789BCD | Registreerimistunnistuse number: | EL601511 |
| Registreerimismärgi tüüp: | A1 – Üldkasutatav auto | Registreerimistunnistuse väljastamise kuupäev: | 01.10.2019 |
| ID-kood: | X8091020061 | Esmase registreerimise kuupäev: | 20.10.2009 |
| VIN-kood (tehasetähis): | TMBDA612A2048332 | Registreerimistunnistuse väljastaja andmed: | Eesti Riiklik Autoregistrikeskus |
| Sõiduki andmed | | Sõiduki omanik | |
| Kategooria: | M1 | Isiku- või äriregistri kood: | 47003280318 |
| Mark: | SKODA | Nimi või perekonnanimi: | MAANTEE-ÕIS-LEHEMETS JULIA |
| Mudel: | OCTAVIA | Elu- või asukoht: | Eesti, Ida-Muru maakond, Kivõli linn, Kuuse tänav 35 |
| Kere tüüp: | LUUKPÄRA | Asukoha kood: | 0309 |
| Mootori tüüp: | BENSIIN_KATALYSAATOR | Muud andmed: | KEHTIV |
| Mootori maht: | 1595 | Sõiduki kasutaja | |
| Mootori võimsus: | 75 | | |
| Ehitusaasta: | 2009 | | |
| Värvus: | MUST | | |
| Registreerimismärk: | 275ADP | Registreerimistunnistuse number: | EL584827 |
| Registreerimismärgi tüüp: | A1 – Üldkasutatav auto | Registreerimistunnistuse väljastamise kuupäev: | 13.08.2019 |
| ID-kood: | A230107045 | Esmase registreerimise kuupäev: | 01.01.1988 |
| VIN-kood (tehasetähis): | W0L000016J1016722 | Registreerimistunnistuse väljastaja andmed: | Eesti Riiklik Autoregistrikeskus |

| | |
|-----------------------|---|
| Sõiduki andmed | Sõiduki omanik |
| Kategooria: MI | Isiku- või äriregistri kood: 47003280318 |
| Mark: OPEL | Nimi või perekonnanimi: MAANTEE-ÕIS-LEHEMETS JULIA |
| Mudel: OMEGA | Blu- või asukoht: Saare maakond, Saaremaa vald, Aru küla, Põldmaa |
| Kere tüüp: SEDAN | Asukoha kood: 1364 |
| Mootori tüüp: BENSIIN | Muud andmed: KEHTIV |
| Mootori maht: 1998 | Sõiduki kasutaja |
| Mootori võimsus: 85 | KASUTAJA: KÜLLI MAANTEE (48004300330); |
| Ehitusaasta: 1988 | |
| Värvus: VALGE | |

| | |
|--|--|
| Registreerimismärk: 597BRB | Registreerimistunnistuse number: EK754692 |
| Registreerimismärgi tüüp: A1 – Üldkasutatav auto | Registreerimistunnistuse väljastamise kuupäev: 11.05.2016 |
| ID-kood: O9160511002 | Esmase registreerimise kuupäev: 10.02.2004 |
| VIN-kood (tehasetähis): VF33ERHYB83342387 | Registreerimistunnistuse väljastaja andmed: Eesti Riiklik Autoregistrikeskus |

| | |
|-----------------------|--|
| Sõiduki andmed | Sõiduki omanik |
| Kategooria: MI | Isiku- või äriregistri kood: 47003280318 |
| Mark: PEUGEOT | Nimi või perekonnanimi: MAANTEE-ÕIS-LEHEMETS JULIA |
| Mudel: 307 | Blu- või asukoht: Eesti, Põlva maakond, Põlva vald, Põlva linn, Nurme tänav 7b-5 |
| Kere tüüp: UNIVERSAAL | Asukoha kood: 6536 |
| Mootori tüüp: DIESEL | Muud andmed: KEHTIV |
| Mootori maht: 1997 | Sõiduki kasutaja |
| Mootori võimsus: 66 | KASUTAJA: MARTIN ALLMAA(38004300013); KASUTAJA: VLADIMIR POPOV(48612156510); |
| Ehitusaasta: 2004 | |
| Värvus: HALL | |

| | |
|--|--|
| Registreerimismärk: 001AMJ | Registreerimistunnistuse number: EL601507 |
| Registreerimismärgi tüüp: A1 – Üldkasutatav auto | Registreerimistunnistuse väljastamise kuupäev: 14.08.2019 |
| ID-kood: I1940929014 | Esmase registreerimise kuupäev: 01.01.1982 |
| VIN-kood (tehasetähis): XTA212100C0239693 | Registreerimistunnistuse väljastaja andmed: Eesti Riiklik Autoregistrikeskus |

| | |
|-----------------------|---|
| Sõiduki andmed | Sõiduki omanik |
| Kategooria: MI | Isiku- või äriregistri kood: 47003280318 |
| Mark: VAZ | Nimi või perekonnanimi: MAANTEE-ÕIS-LEHEMETS JULIA |
| Mudel: 2121 | Blu- või asukoht: Saare maakond, Saaremaa vald, Aru küla, Põldmaa |
| Kere tüüp: LUUKPÄRA | Asukoha kood: 1364 |
| Mootori tüüp: BENSIIN | Muud andmed: KEHTIV |
| Mootori maht: 1580 | Sõiduki kasutaja |
| Mootori võimsus: 59 | |
| Ehitusaasta: 1982 | |
| Värvus: TUMEPUNANE | |

| | |
|--|--|
| Registreerimismärk: 700DBB | Registreerimistunnistuse number: EL592926 |
| Registreerimismärgi tüüp: A1 – Üldkasutatav auto | Registreerimistunnistuse väljastamise kuupäev: 25.02.2019 |
| ID-kood: A9180316044 | Esmase registreerimise kuupäev: 05.04.2017 |
| VIN-kood (tehasetähis): WBS3R91080K800080 | Registreerimistunnistuse väljastaja andmed: Eesti Riiklik Autoregistrikeskus |

| | |
|------------------------------------|---|
| Sõiduki andmed | Sõiduki omanik |
| Kategooria: MI | Isiku- või äriregistri kood: 47003280318 |
| Mark: BMW | Nimi või perekonnanimi: MAANTEE-ÕIS-LEHEMETS JULIA |
| Mudel: M1 | Blu- või asukoht: Saare maakond, Saaremaa vald, Aru küla, Põldmaa |
| Kere tüüp: KUPEE | Asukoha kood: 1364 |
| Mootori tüüp: BENSIIN_KATALYSAATOR | Muud andmed: KEHTIV |
| Mootori maht: 2979 | Sõiduki kasutaja |
| Mootori võimsus: 317 | |
| Ehitusaasta: 2017 | |
| Värvus: HALL | |

| | |
|--|--|
| Registreerimismärk: 865M-LJ | Registreerimistunnistuse number: EK516274 |
| Registreerimismärgi tüüp: A1 – Üldkasutatav auto | Registreerimistunnistuse väljastamise kuupäev: 16.02.2015 |
| ID-kood: X8120411125 | Esmase registreerimise kuupäev: 11.04.2012 |
| VIN-kood (tehasetähis): SB1BK76LXE027145 | Registreerimistunnistuse väljastaja andmed: Eesti Riiklik Autoregistrikeskus |