

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond  
Informaatikainstituut

Sander Sepp 142427IABB

# **ELEKTRILEVI LIITUMISVORMIDE ÄRIANALÜÜSI TEHNILINE LAHENDUS**

Bakalaureusetöö

Juhendaja: Maili Markvardt  
MSc

Tallinn 2017

## **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: Sander Sepp

22.05.2017

## **Annotatsioon**

Töö põhieesmärgiks on täiendada Elektrilevi iseteeninduskeskkonda uuendades ning juurde luues liitumisvorme. Täidetud vormid tuleb edastada uute teenustega liitumiste infosüsteemile.

Töös räägib autor projektis kasutuselolevatest tehnoloogiatest ning kirjutab liitumisvormidest detailideni. Räägitakse, millistest osadest peavad vormid koosnema, milliseid arhitektuurilisi otsuseid tehti, et luua taotlused, mis jagavad suurt ühisosa. Töös kirjeldatakse lugejale arendusprotsessi Liferay, Java ning Springiga. Lahendus peab optimiseerima liitumisspetsialistide tegevust ning võimaldama kliendile lihtsamat ning ülevaatlikumat taotluse esitamist.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 36 leheküljel, 3 peatükki ja 42 joonist.

## **Abstract**

### Technical implementation for business analysis of Elektrilevi application forms

The main purpose of this work is to update and create application forms for Elektrilevi E-Service. Forms must be sent to new application information system with new services.

Author of the thesis talks about used technologies and writes about forms in detail. Also which sections must be in different forms, what architectural decisions were made to create forms that share a lot in common. Thesis describes developing process with Liferay, Java and Spring. Solution must optimize application specialist's work and provide better overview of the application process to the customer.

The thesis is in estonian and contains 36 pages of text, 3 chapters and 42 figures tables.

## Lühendite ja mõistete sõnastik

ELV	Elektrilevi OÜ
KHO	ELV Kliendihalduse osakond
Liitumisspetsialist	ELV KHO liitumise protsessi juhtija
Liitumine	Liitujale uue võrguühenduse loomine, olemasoleval võrguühendusel liitumispunkti kaitsme nimivoolu suurendamine madalpingel või tarbimisvõimsuse suurendamine keskpingel
Taotlus	Dokument, millega kirjeldatakse kliendi soov ELV poolt pakutava toote saamiseks
KMP	Kaugus (kuni 400m) olemasolevast ELV omandis või kasutuses olevast keskpinge alajaamast
SOAP	XML-i baasil sõnumite vahetamise protokoll
AXIS	Teenuste genereerimise vahend
CMS	<i>Content management system</i> – sisuhaldus süsteem
RIK	Registrite ja infosüsteemide keskus

## Sisukord

1 Sissejuhatus.....	10
2 Kasutatavate tehnoloogiate kirjeldus .....	11
2.1 Liferay.....	11
2.2 Apache Axis2.....	12
2.3 Spring MVC.....	12
2.4 Unit testing ja TestNG .....	13
2.5 Ivy ja Ant .....	14
2.6 jQuery validation engine.....	14
2.7 Liquibase.....	15
3 Elektrilevi iseteeninduse süsteemi kirjeldus .....	17
4 Ärianalüüsi tehniline lahendus.....	18
4.1 Autentimine, volitused ja rollid .....	19
4.2 Portleti loomine ning konfigureerimine.....	20
4.3 Tõlked .....	22
4.4 Teavitused .....	22
4.5 Ühised sektsioonid .....	24
4.5.1 Kontaktandmed .....	24
4.5.2 Tarbimiskoha sisestus/valik .....	25
4.5.3 Mikrotootmisega seotud andmed.....	29
4.5.4 Dokumendid.....	30
4.5.5 Tingimustega nõustumine.....	32
4.6 Uus liitumine madalpingel.....	32
4.6.1 Liitumispunkti parameetrid.....	33
4.6.2 Liitumispunkti asukoht .....	33
4.6.3 Liitumise orienteeruv hind.....	35
4.7 Tarbimistingimuste muutmine .....	36
4.7.1 Tooted .....	37
4.7.2 Hinnaarvutus .....	38
4.8 Elektrivõrgu ümberehitus.....	39

4.9 Mikrotootja liitumine .....	40
4.9.1 Võrguühenduse andmed.....	40
4.9.2 Mikrotootmisega seotud andmed .....	40
4.10 Tehnilised tingimused .....	42
4.10.1 Tasuta tingimused .....	43
4.10.2 Tasulised tingimused .....	44
4.11 Testimine.....	45
5 Kokkuvõte.....	47
Kasutatud kirjandus .....	49

## Jooniste loetelu

Joonis 1. Springi päringutöötlemise struktuur .....	13
Joonis 2. Vana veakuvamise stiil .....	15
Joonis 3. Uue veakuvamise stiil .....	15
Joonis 4. Liquibase'i <i>changelog</i> 'i <i>changeset</i> .....	16
Joonis 5. Nõrga volituse infotekst .....	20
Joonis 6. Portleti konfiguratsioon .....	20
Joonis 7. Portleti kontekst .....	21
Joonis 8. Portleti kaasatavad Javascripti failid .....	21
Joonis 9. Liitumisvormide grupeering admini vaate jaoks .....	22
Joonis 10. Pakkumise teavitus .....	24
Joonis 11. <i>PhoneType enum</i> klass .....	25
Joonis 12. Kontaktandmete sektsioon .....	25
Joonis 13. Tarbimiskoha aadressi sisestamise sektsioon .....	26
Joonis 14. Uue aadressi sisestamine .....	27
Joonis 15. Ühe tarbimiskoha kuva .....	28
Joonis 16. 2-5 tarbimiskohta kuva .....	28
Joonis 17. Üle 5 tarbimiskohaga sisestusväli .....	28
Joonis 18. Infoteade korteri valimisel .....	29
Joonis 19. Mikrotootmisega seotud andmete sektsioon .....	29
Joonis 20. Dokumentide sektsioon .....	31
Joonis 21. Veateade vale laiendiga faili sisestamisel .....	31
Joonis 22. Mahu ületamisel kuvatav veateade .....	31
Joonis 23. Tingimustega nõustumise sektsioon .....	32
Joonis 24. Liitumispunkti parameetrid .....	33
Joonis 25. Tarbimiskoha valideerimine .....	34
Joonis 26. Liitumispunkti asukoht .....	35
Joonis 27. Orienteeruva hinna kuva .....	36
Joonis 28. Tegelike kulude info kuva .....	36



Joonis 29. Peakaitsme suurendamise, liitumispunkti asukoha muutmise ja faaside suurendamise tooted.....	38
Joonis 30. Võrguühenduse andmed .....	40
Joonis 31. Tarbimiskoha mikrotootmise info .....	41
Joonis 32. Peakaitsme suurendamine ja faaside muutmine .....	42
Joonis 33. 3x220V liinipingelt üleminek 3x400V pingele .....	43
Joonis 34. Liitumine keskpingel .....	43
Joonis 35. Tehnilised tingimused hoone elektriprojekti koostamiseks.....	44
Joonis 36. Liinimastide ühiskasutus .....	44
Joonis 37. Mitteelektriprojekti koostamine.....	44
Joonis 38. Detailplaneeringu võimsuse sektsioon .....	45
Joonis 39. Detailplaneeringu peakaitsme sektsioon.....	45
Joonis 40. Ajutise võrguühenduse võimsuse sektsioon .....	45
Joonis 41. Ajutise võrguühenduse peakaitsme sektsioon .....	45
Joonis 42. Näide telefoninumbri testist.....	46

## 1 Sissejuhatus

Elektrilevi üheks peamiseks töövooks on võrgulepingute sõlmimine ning nendega erinevate toimingute tegemine. Kasutaja saab esitada iseteeninduses taotlusi, mis edastatakse liitumisspetsialistidele, kes töötlevad saadud informatsiooni ning viivad läbi toiminguid, mis võimaldavad kliendil saada omale vajaliku. Praegune taotluse esitamise elutsüklil on raskesti hallatav nii liitumisspetsialistidele kui ka klientidele endile – puudub ühtne ülevaade toimuvast.

Taotluste mugavamaks esitamiseks ning haldamiseks on vajalik luua süsteem Elektrilevi iseteeninduse ning liitumiste infosüsteemi vahel. Lõputöö eesmärk on luua 2 uut ning muuta 3 olemasolevat taotlusvormi. Lisaks tehakse vaade, kus autentimata lehe külastaja saab võimaluse hinnata oma ligikaudseid kulutusi uue liitumise korral. Kaks täiesti uut vormi on tehnilised tingimused ja mikrotootja liitumine. Viimaseid saab hetkel esitada vaid paberi peal. Uue liitumise, tarbimistingimuste muutmise ning elektrivõrgu ümberehituse vormidele luuakse juurde funktsionaalsust ning lisatakse kliendile olulist infot, näiteks ligikaudse hinnaarvutuse näol. Täiendada tuleb ka Elektrilevi iseteeninduse teavituste funktsionaalsust, et klienti oleks võimalik koheselt teavitada, kui pakkumus esitatud taotlusele on tehtud.

Töö koosneb 3 peatükist. Esimeses peatükis tuuakse välja peamised tehnoloogiad, mida töö tegemisel kasutati, mille jaoks neid vaja oli ning olulisemate juurde on selgitatud ka nende töö põhimõtet. Teises peatükis räägitakse Elektrilevi iseteeninduse olemusest ning põhjusest, milleks ta üldse loodi. Kolmandas peatükis kirjeldatakse lähemalt projekti üldist struktuuri, loodavate liitumisvormide osasid ja nende tööks vajalikku loogikat. Samuti räägitakse koodi ning kasutajaliidese testimisest.

## 2 Kasutatavate tehnoloogiate kirjeldus

Elektrilevi iseteenindust on arendatud juba mitmeid aastaid, seetõttu olid keskkond ning tehnoloogiad enne lõputöö alustamist juba paigas. Arvestades iseteeninduse vanust, pole kasutatavad tehnoloogiad kõige tänapäevasemad, kuid kes ei tea minevikku, elab tulevikuta.

### 2.1 Liferay

Liferay on üks levinumaid Java põhiseid CMS-e (*content management system*), mis kujutab endast robustset platvormi ning võimaldab ettevõttel kiiresti üles ehitada oma portaali. Temaga on kaasas põhilised rakendused, mis on tavaliselt veebilehtedel alati kasutuses. Samuti on Liferay'1 lihtsasti kasutatav arendusraamistik, mis teeb uute rakenduste ehitamise võrdlemisi lihtsaks. (Sezov, 2016)

Liferay kasutab portletstruktuuri. Portlet on veebilehe komponent, mis koosneb visuaalsest poolest ning funktsionaalsusest, mis on implementeeritud Javas. Iga lehele saab lisada mitmeid portlete ning hallata nende konfiguratsiooni ja kasutaja õigusi individuaalselt. (Ozawa, 2011)

Liferay seadistamiseks tuleb luua tühi andmebaas ning valida vaikimisi tähtede kodeeringuks UTF-8, sest Liferay on mitmekeelne rakendus. Seejärel tuleb luua andmebaasi kasutaja, kellel oleks õigused tabelite loomiseks ning kustutamiseks. Hiljem tuleb kasutada loodud kasutajat ühendamiseks Liferay rakendust andmebaasiga. Esmakordselt Liferay-d käivitades luuakse andmebaasi kõik vajalikud tabelid koos indeksitega. Oluline on seda kõike teha eelnevalt loodud andmebaasi kasutajaga, sest nii oskab Liferay hiljem pistikprogrammide või millegi taolise lisamisel luua uusi tabeleid. (Sezov, 2016)

## 2.2 Apache Axis2

Elektrilevi veebikeskkond on juba suhteliselt ammu arendatud, mistõttu on seal kasutuses REST teenuste asemel SOAP-i teenused. Teenuste genereerimiseks on projektis kasutusel Apache Axis2.

Apache Axis2 tutvustati esmakordselt 2004 aasta augustis ning omab täiesti uut arhitektuuri võrreldes oma eelkäijaga, Apache Axisega. Axis2 on võimekam, modulaarsem ning rohkem XML-le orienteeritud, kui tema eelkäija. Süsteem on disainitud nii, et oleks võimalikult lihtne ühendada temaga teisi pistikprogramme, näiteks WS-Security, et tagada suurem turvalisus sõnumite vahetamisel. (Why Apache Axis2, 2016)

## 2.3 Spring MVC

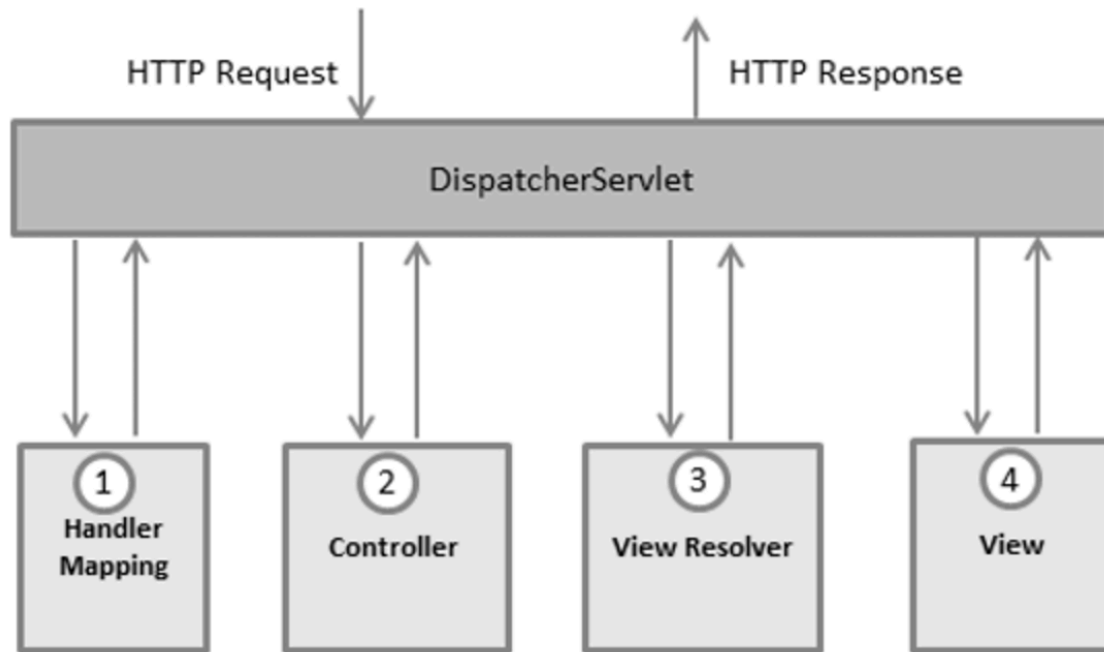
Spring MVC võimaldab arendada veebirakendusi *model-view-controller* arhitektuuriga, mis on paindlikud ning selle komponendid on oma vahel vähe seotud. MVC arendusmuster võimaldab lahus hoida äri loogika, kasutajaliidese loogika ning rakenduse osade kasutamise.

Model – hõlmab endas andmestruktuuri ehk sisuliselt POJO (*plain old java object*).

View – vaatele antakse ette model ehk info, mille alusel genereeritakse kliendile veebileht.

Controller – peamine ülesanne on käsitleda sissetulevaid päringuid ning ette valmistada andmemudel, mis edastada genereerimaks vaade.

Spring MVC üheks peamiseks komponendiks on DispatcherServlet, mis käsitleb kõiki sissetulevaid päringuid ning nende vastuseid.



Joonis 1. Springi päringutöötlemise struktuur

Pärast HTTP päringu sissetulemist otsustavad DispatcherServlet ning HandlerMapping, millise *controlleri* poole pöörduda. Seejärel kutsub *controller* välja vastava meetodi, millele GET või POST päring tehti, ning edastab DispatcherServletile vaate nime, kes omakorda annab info edasi ViewResolverile. Lõpuks paneb DispatcherServlet vaatele modeli kaasa, mille põhjal saab vajaliku lehe valmis genereerida. (Spring MVC Framework)

## 2.4 Unit testing ja TestNG

TestNG on testimisraamistik, mis on inspiratsiooni saanud JUnitilt ja NUnitilt ning omaltpoolt lisab juurde uusi funktsionaalsusi, mis muudavad raamistiku võimsamaks ning lihtsamalt kasutatavaks. (Beust, 2015)

Ühiktestimise eesmärk ei ole leida ainult koodivigasid. Tõepoolest, kui koodi kirjutatakse ümber ehk refaktoreeritakse, siis ühiktestid toovad välja võimalikud rakenduse käitumiserinevused. Siiski, ühiktestimise peamiseks eesmärgiks on tarkvarakomponentide disainimine võimalikult robustselt. Seeläbi on võimalik koodi kirjutada nii, et iga meetod või ühik koodis täidaks ühte konkreetset ülesannet, samuti võimaldab ära määrata meetodi sisendid. Sisenditele mõeldes on võimalik koodi disainida nii, et testimisel oleks võimalikult vähe vaja kasutada *mock*'imist. Testide

haldamise vaatenurgast on oluline ka viimaste nimetamine – peaks olema mainitud testitava meetodi nimi ning spetsiifilise olukorra oodatav tulemus. (Sanderson, 2009)

## 2.5 Ivy ja Ant

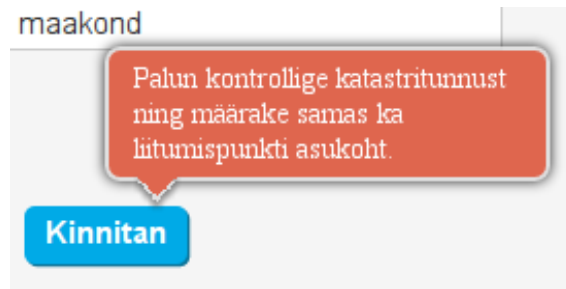
Ivy on vabavaraline vahend käsitlemaks projekti sõltuvusi ning on hästi integreeritav Apache Ant-iga. Ant on Java teek ja käsurea vahend, millega on võimalik kirjutada projekti vajalikke käskusid ehk *targeteid*. Talle on sisseehitatud ka peamised käsud projekti koodi kompileerimiseks, ehitamiseks ning testide jooksumiseks. *Target*'eid on võimalik üksteisega ka siduda, see tähendab, et üks käsk kutsub välja teised käsud. (Apache Ant TM)

Antud projektis on ise arendatud Ant'i käsud, mis võimaldavad kogu projekti esmakordselt ülesse seada. Näiteks on kirjutatud *target*'id, mis impordivad vajalikud SSL sertifikaadid Tomcat'i, seavad ülesse Liferay ning MySQL andmebaasi.

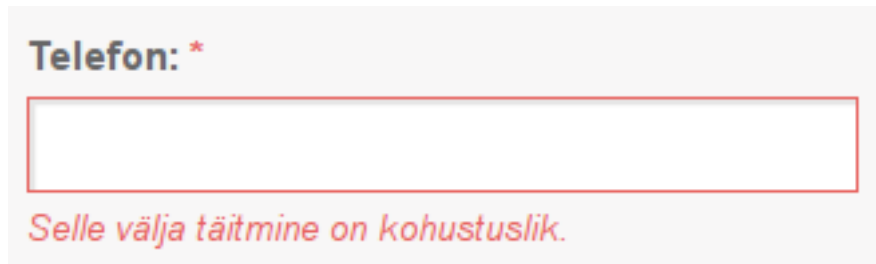
## 2.6 jQuery Validation Engine

jQuery validation engine on Javascripti pistikprogramm, mille abil on võimalik hõlpsasti valideerida vormi väljasid. Temaga tuleb kaasa olulisel hulgal sisseehitatud valideerimisreegleid, näiteks e-meil, telefon, URL jpt. *Plug-in*'i koodi on võimalik kirjutada juurde ka enda reegleid, mis järgivad etteantud regulaaravaldist. Lisaks on võimalik kirjutada välja kontrole ka Javascript'i, mis muudab validatsioonide tegemise palju paindlikumaks – pole vaja kõike kirjutada lähtekoodi. (Dugas, 2016)

Erinevate validatsioonivigade jaoks on võimalik deklareerida veateadete tekstid. Neid saab kirjutada nii lähekoodi kui ka initsialiseerida Javascript'is. Vaikimisi on *validation engine*'l endal veakuvamisstiil (Joonis 2). Seoses Elektrilevi ning Eesti Energia uute arendustega, muutusid ka veateadete kuvamise stiilid (Joonis 3). Seetõttu oli vaja modifitseerida pistikprogrammi vaikimisi käitumist.



Joonis 2. Vana veakuvamise stiil



Joonis 3. Uue veakuvamise stiil

## 2.7 Liquibase

Liquibase on avatud lähtekoodiga andmebaasi muudatuste haldamise vahend, mis on ehitatud Javas. Liquibase'ga ei pea kirjutama SQL-i otse andmebaasi, et luua, uuendada või kustutada andmebaasist objekte. See-eest kirjutatakse muudatused xml failidesse ehk *changelogidesse*. *Changelogides* on omakorda muudatused jaotatud gruppidesse ehk *changesettidesse*. Liquibase'i saab käivitada näiteks käsurealt või integreerida hoopis Ant'i või Maven'iga.

Liquibase'i käivitamiseks tuleb esmalt ära määrata andmebaas, kuhu muudatused tehakse. Liquibase ise kasutab kahte tabelit haldamaks andmebaasi tehtud muudatusi: DATABASECHANGELOG ja DATABASECHANGELOGLOCK. Kui tabeleid andmebaasis ei eksisteeri, siis nad luuakse. DATABASECHANGELOGLOCK tabelit kasutatakse selleks, et ainult üks Liquibase'i instants oleks töös. DATABASECHANGELOG tablis on kirjas kõik *changesetid*, mis on Liquibase poolt käivitatud. (McGarr, 2010) Igal *changesetil* on oma id ja autori nimi ning tema sisse saab kirjutada nii tavalist SQL-i kui ka eelnevalt defineeritud andmebaasi protseduure (Joonis 4). Jooniselt on näha kahte protseduuri – eelnevalt loodud tõlke kustutamine ning uue lisamine.

```
<changeSet id="portlets.agreements.04.05.2017.2" author="Sander Sepp" runOnChange="false">
  <sql>
    <![CDATA[
      call ee_delete_translation('portlets.agreements.networkagreementdetails.pricePerYear');
      call ee_insert_translation_all('portlets.agreements.networkagreementdetails.pricePerYear', 'Võrgutasu orienteeruv hind aastas');
    ]]>
  </sql>
</changeSet>
```

Joonis 4. Liquibase'i *changelog*'i *changeset*



### **3 Elektrilevi iseteeninduse süsteemi kirjeldus**

Iseteeninduse projekti eesmärgiks oli luua Eesti Energiale järgmise põlvkonna e-äri lahendus ning ühtlustada kõigi kontserni ettevõtete välisveebide visuaalne ning ülesehituslik kontseptsioon. (Tint, 2010)

Iseteenindus on peamiseks kanaliks kliendiga suhtlemisel. Tehniliselt tähendab see suurt kasutajate hulka, kellele tuleb pakkuda mugavat ning kiiret kasutajaliidest erinevate funktsionaalsustega. Eesti Energias on palju erinevaid süsteeme, millega antud süsteem liidestub. Näiteks näidu teatamise taga asub mõõteinfosüsteem, kliendid paiknevad kliendiinfosüsteemis. Lõputöö raames luuakse ka liidestus liitumiste infosüsteemiga. Enamasti ei tiritu süsteemide andmeid iseteenindusse alla – andmevahetus teiste süsteemidega käib veebiteenuste kaudu. Siiski salvestatakse kliendi primaarandmed ja teatud tarbimiskohtade info sessiooni. (Metsis, 2013)

## 4 Ärianalüüsi tehniline lahendus

Nii Eesti Energia kui ka Elektrilevi iseteenindus arendati välja juba 2011 aastal ning ajapikku on sinna arendatud palju uut ning täiendatud vana funktsionaalsust, mis tähendab, et koodibaas, milles orienteeruda, on päris mahukas. Kuna kasutuses on Liferay CMS, siis peamised veebilehe komponendid, mida klient oma tegemisteks kasutab, on portletid. Teatud osad lehtedel on kirjutatud ka lihtsalt veebisisuna. Veebisisu on tavaline tekst, kuid võib olla ka HTML-s kirjutatud koodijupikene. Registreerimine, kontaktandmete uuendamine, sisse logimine, arvete maksmine, lepingute manageerimine – need funktsionaalsused moodustavad marginaalse osa kogu veebilehest, mistõttu tuleb portletide failid loogiliselt ära paigutada, et kood ei muutuks hallatamatuks.

Lõputöö käigus tuleb luua 6 vormi, millest 5 sisu saadetakse liitumiste infosüsteemi, 1 neist on mõeldud autentimata kasutajale, kes saab uurida infot oma maa ning liitumispunkti kohta. Kasutajaliidesed on täiesti mahus loodud Eesti Energia ja Elektrilevi disainerite poolt. Kuna loodava funktsionaalsuse hulk on päris suur, tuli enne arendama hakkamist põhjalikult läbi mõelda süsteemi arhitektuur. Põhimõtteliselt oleks saanud kogu funktsionaalsust arendada ka ühes portletis, kuid koodi loetavuse ja hallatavuse vaatenurgast poleks see kuigi mõistlik olnud. Vaadates lähemalt sisendina ette antud süsteemianalüüsi, jagasid vormid päris mitmeid ühiseid osasid – teatud sektsioonid olid igas, teised aga 1-2 vormis olemas. Tuginedes eelnevale, oli võimalik arhitektuuri loogiliselt ja loetavalt paika panna. Lõpptulemuseks oli 6 portleti.

Võttes arvesse fakti, et kõik vormid jagavad omavahel teatavat ühisosa, loodi üks abstraktne kontrolleri, mille ülesandeks on vaatesse kaasa anda info, mis on olemas igas vormis. Seega iga vormi spetsiifiline kontrolleri pärineb abstraktsest kontrolleri, milles on defineeritud ka abstraktsed meetodid, mis tuleb implementeerida iga vormi spetsiifilises kontrolleri. Nendeks meetoditeks on `fillModel`, mille ülesandeks on spetsiifilise info modelisse panemine ning `absView`, mis tagastab portleti vaate asukoha failisüsteemis. Abstraktses kontrolleri on defineeritud ka meetod, kuhu

jõutakse alati esimesena ning viimasena. See on meetod, mis valmistab ette üldise infomudeli, suunab programmi vastavasse kontrollerrisse, millest tuli päring, ning lõpuks tagastab õige vaate.

Projektis on üldine tava see, et kontrollerr hoolitseb päringute vastu võtmisega ning äriloogika on implementeeritud *helperites*. See tagab koodi modulaarsuse ning taaskasutatavuse erinevate komponentide vahel. Erinevate *helperite* kaasamine

Vormide ühise osa koos hoidmiseks loodi projekti *common* pakett, mis hõlmab endas abstraktset kontrollerrit, ühiseid *helpereid*, komparaatoreid ning tavalisi andmeobjekte, mis aitavad teenustest saadud andmeid vaate põhiselt paremini strukturiseerida. Komparaatoreid on vaja luua seepärast, et projektis kasutusel olev Liferay versioon ei toeta Java 8-t, kus on andmete sorteerimine ning töötlemine palju mugavamaks tehtud tänu *streamidele*.

## 4.1 Autentimine, volitused ja rollid

Lõputöö raames on olulisel kohal autentimine ning iseteeninduses kasutajatele antud volitused. Tugev autent saavutatakse sisselogimisel kas ID-kaardi, mobiili-ID või panga kaudu. Nõrk autent saadakse sisselogimisel viitenumbri ja ning sellisel juhul ei ole kasutajal õigus teha iseteeninduses enamikke toiminguid ja vajadusel palutakse sisse logida kasutades mõnda eelpool mainitud tugevat autentimisviisi. Olemas on 4 tüüpi volitusi: peakasutaja, teise kasutaja täisvolitus, kontopõhine volitus, millega saab maksta ning volitus, millega saab lepingut lihtsalt vaadata. Kasutajal võib olla mitu kontot, näiteks on tal mitu tarbimiskohta ning igal tarbimiskohal on oma konto.

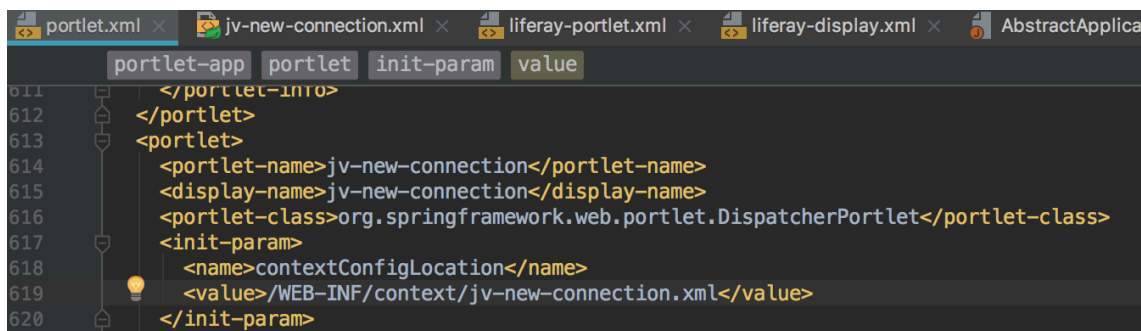
Uue liitumise, mikrotootja liitumise ning tarbimistingimuste muutmise korral on oluline, et klient oleks enda rollis ehk peakasutaja või omab teise kasutaja täisvolitust ehk talle on antud õigus täielikult sõlmida ning muuta lepinguid. Vastasel korral kuvatakse vormi asemel nõrga volituse infoteadet (Joonis 5). Tehniliste tingimuste ning elektrivõrgu ümberehituse korral ei ole nõutud peakasutaja või täisvolituse õigusi, piisab lihtsalt tugevast autendist.

Teenuste tellimiseks on vajalik kõrgem volituse tase.

Joonis 5. Nõrga volituse infotekst

## 4.2 Portleti loomine ning konfigureerimine

Portleti tegemiseks tuleb lisada kirjed mitmesse xml faili. Esimeseks failiks on portlet.xml, kus saab määrata portleti nime ning portleti klassi. Antud lõputöö raames loodud portletid kasutavad klassi DispatcherPortlet (Joonis 6).



```
611 </portlet-info>
612 </portlet>
613 <portlet>
614   <portlet-name>jv-new-connection</portlet-name>
615   <display-name>jv-new-connection</display-name>
616   <portlet-class>org.springframework.web.portlet.DispatcherPortlet</portlet-class>
617   <init-param>
618     <name>contextConfigLocation</name>
619     <value>/WEB-INF/context/jv-new-connection.xml</value>
620   </init-param>
```

Joonis 6. Portleti konfiguratsioon

DispatcherPortlet võtab vastu veebist tulevaid päringuid ning juhatab nad õigestesse kontrolleri meetoditesse. Igal DispatcherPortletil on oma WebApplicationContext, mis pärineb globaalsest WebApplicationContextist. DispatcherPortleti initsialiseerimisel otsitakse WEB-INF kataloogist faili nimega “portleti-nimi”-portlet.xml, kus on ära määratud bean-id ehk oad. Viimati mainitud fail määrab seega ära portleti konteksti ehk koodis kasutatavad objektid. (The DispatcherPortlet)

Edasi määratakse ära vajalikud parameetrid. Antud projektis on vaikimisi konteksti lugemise koht ära muudetud – ube ei otsita mitte WEB-INF kataloogist, vaid WEB-INF/context kataloogist, mis on tehtud selleks, et luua failide hoidmiseks parem struktuur (Joonis 7). ContextConfigLocation parameeter määrab ära, millisest failist minnakse konteksti lugema.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://www.springframework.org/schema/beans
5     http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
6
7     <import resource="classpath:/portlet-processing.xml"/>
8
9     <bean id="newConnectionController" class="ee.energia.jvapplication.newconnection.NewConnectionController"/>
10    <bean id="newConnectionHelper" class="ee.energia.jvapplication.newconnection.NewConnectionHelper"/>
11    <bean id="contactInfoHelper" class="ee.energia.jvapplication.common.ContactInfoHelper"/>
12    <bean id="applicationHelper" class="ee.energia.jvapplication.common.ApplicationHelper"/>
13    <bean id="cadastreHelper" class="ee.energia.jvapplication.newconnection.CadastreHelper"/>
14    <bean id="portletMultipartResolver" class="org.springframework.web.portlet.multipart.CommonsPortletMultipartResolver">
15        <property name="maxUploadSize" value="5242880" />
16    </bean>
17 </beans>
```

Joonis 7. Portleti kontekst

Portlet.xml failis on ka ära määratud portletid, mis läbivad AccountsRequiredFiltrit. See filter määrab rangelt ära, et lehe külastaja, peab olema autenditud. Vastasel juhul keelatakse igasugused toimingud portletiga ning sisu välja ei kuvata. Ainus portlet, mis antud töös AccountsRequiredFilter't ei vaja, on võrguühenduse info vaatamiseks, kus seost kasutajaga pole.

Järgmise oluline fail on liferay-portlet.xml, kus on ära määratud portleti spetsiifilised *javascript*-i ning *css*-i failid ning koht kus neid kaasata. Näiteks saab kaasata faile nii DOM-i päisesse kui ka jalusesse (*header-portal-javascript* või *footer-portal-javascript*). Kuna portleti vaade genereeritakse DOM-i sisse, siis on võimalus *script*-i faile kaasata ka vahetult enne või pärast portlet-i html koodi (*header-portlet-javascript* või *footer-portlet-javascript*) (Joonis 8).

```
<portlet>
  <portlet-name>jv-new-connection</portlet-name>
  <header-portal-javascript>/html/js/general/proj4js-combined.js</header-portal-javascript>
  <header-portlet-javascript>/js/jv-application/jv-validate.js</header-portlet-javascript>
  <footer-portal-javascript>/js/jv-application/jv-application.js</footer-portal-javascript>
  <footer-portlet-javascript>/js/jv-application/jv-new-connection.js</footer-portlet-javascript>
  <footer-portlet-javascript>/js/jv-application/form.js</footer-portlet-javascript>
</portlet>
```

Joonis 8. Portleti kaasatavad Javascripti failid

Viimaseks failiks on liferay-display.xml, kus saab luua portletide jaoks kategooria ning nad sinna id põhiselt lisada. Seal deklareeritud kategooriad ja *portletid* kuvatakse välja Liferay admini vaates (Joonis 9).

```
<category name="portlets.jvapplications">
  <portlet id="jv-application-details"/>
  <portlet id="jv-application-list"/>
  <portlet id="jv-new-connection"/>
  <portlet id="jv-new-connection-anonymous"/>
  <portlet id="jv-change-parameters"/>
  <portlet id="jv-rebuild"/>
  <portlet id="jv-micro-producer"/>
  <portlet id="jv-technical-inspection"/>
</category>
```

Joonis 9. Liitumisvormide grupeering admini vaate jaoks

### 4.3 Tõlked

Elektrilevi iseteeninduskeskkonda on võimalik vaadata kolmes keeles – eesti, inglise ja vene keeles. Antud projektis on kasutusel Liquibase, mida kasutatakse tõlgete ning Liferay muutujate lisamiseks andmebaasi. SQL Lausete mugavamaks kirjutamiseks ning lugemiseks on loodud andmebaasi kohandatud protseduurid: `ee_insert_translation_all`, `ee_delete_variable`, `ee_insert_variable` ja `ee_delete_variable`. `Ee_insert_translation_all` võtab sisendiks tõlke id ning teksti ja genereerib andmebaasi igas keeles sisestatud tõlke. Näiteks on eesti keeles sõna “Kontaktandmed”, siis vene keeles on “Kontaktandmed (RU)” ja inglise keeles “Kontaktandmed (EN). Seeläbi on võimalik testijatel kinnitada, et mitmekeelsus on realiseeritud. Reaalsed tõlked luuakse Elektrilevi äripoole poolt ning neid saab muuta Liferay-st või sisse lugeda Exceli failist.

### 4.4 Teavitused

Lisaks 6-le vormile oli vaja täiendada ka teavituste funktsionaalsust – kliendile tuleb kuvada teavitus esitatud taotlusele tehtud pakkumisest. Enne lõputööd oli kogu teavituste osa täielikult seotud kliendiinfosüsteemiga, kus genereeriti vastavates olukordades teavitus, mis hõlmas endas kindlat infot, mille põhjal suudeti Elektrilevi iseteeninduses valmis genereerida *notification*. Iseteenindusse on implemeteeritud “töö”, mis käivitatakse iga 5 minuti tagant – käiakse küsimas kliendiinfosüsteemist uusi teavitusi. Kuna lõputöö raames loodavad vormid saadavad oma info liitumiste infosüsteemi, mis pole kuidagi seotud klieniinfosüsteemiga, siis tuligi vana implementatsiooni uuendada. Sarnaselt Liferay-ga, kus kogu CMS-i konfiguratsioon ja sisu hoitakse kohalikus MySQL andmebaasis, nii hoitakse ka teavitusi kohalikus andmebaasis.

Üheks äriliseks nõudeks oli võimalus sisse ja välja lülitada liitumiste infosüsteemi teavitusi. Kui teavitused on lubatud, pannakse käima `GetSubmittedConnectionApplications` teenus, mis tagastab kliendi poolt esitatud taotlused. Selleks, et taotlus tehtaks teavituseks, peab ta vastama järgmistele tingimustele: `contractOfferCreatedDate` ja `contractOfferIssuedDate` peavad olema väärtustatud ning `contractSignedDate`, `applicationCancelledDate` ja `jobCancelledDate` ei tohi olla väärtustatud. See kontroll valideerib, et vastav taotlus on liitumisspetsialistide poolt üle vaadatud või läbinud automaatprotsessi ning temale on genereeritud pakkumus. Samas ei tohi olla leping lõpetatud või allkirjastatud, sest sel juhul pole tegu enam valiitse pakkumusega, vaid hoopis lepinguga, mis on jõudnud oma olekuga hilisemasse protsessifaasi. Automaatprotsessi rakendatakse taotlustele, mis täidavad kindlaksmääratud nõudeid ning lõpptulemusena genereeritakse taotlusele pakkumus ilma, et liitumisspetsialist peaks manuaalselt midagi tegema.

Teavitused on jaotatud kahte suuremasse gruppi: `OFFER` või `GENERAL`, mis on ka tugevamalt seotud kasutajate ning kontode volitustega. Sisseloginud kasutajale võib olla teine kasutaja andnud volituse toimingute tegemiseks ning seda aspekti tuleb järgida ka teavituste puhul. `GENERAL` teavituste grupp seob teavitused vaid sisseloginud kasutajaga, kuid oluline on näha pakkumuse teavitust ka volituse saanud kontodel. Oma olemuselt on taotluse pakkumus `OFFER` ning viimane võimaldab teavitust kuvada ka teistel kontodel.

Teavituste sisselugemisel tuleb väärtustada ka tüüp ning vajalikud parameetrid. Tüübi järgi on Liferay-sse loodud muutujad ning tõlked, mille põhjal vaadet genereerides saadakse kätte kindlad tekstid ja URL, mida teavitusel kuvada. URL-i genereerides antakse talle kaasa teavituse küljes olevad parameetrid, mille põhjal leitakse üles õige lepingu pakkumus.



Joonis 10. Pakkumise teavitus

## 4.5 Ühised sektsioonid

Kõigi 6 vormi vahel on ühisosa ehk sektsioonid, mis korduvad. Programmeerimise üheks olulisemaks reegliks on võimalikult palju ära kasutada juba loodud koodi. Viimase duplitseerimine lõppeb üldjuhul katastroofiga, kuna nii muutub kood üsna peagi haldatamatuks ning iga muudatus muutub üha kulukamaks. Seega loodi koodi struktuur nii, et kõik oleks võimalikult universaalne ning sõltumatu kontekstist.

### 4.5.1 Kontaktandmed

Kontaktandmete sektsioon on esindatud kõigis vormides. Kliendi andmete küsimiseks kliendiinfosüsteemist oli eelnevalt loodud teenus `GetPersonsPrimaryData`, mis nõuab sisendiks ühte või mitut kasutaja id-d ning tagastab kasutaja/te primaarandmed. Antud sektsiooni jaoks olid olulised 5 välja – eesnimi, perekonnanimi, telefon ning e-mail. Kuna kliendil võib olla numbreid rohkem kui üks ning eristatud on 4 eri tüüpi numbreid – mobiil, lauatelefon, välismaa telefon ning faks, siis tuli numbreid sorteerida tüübi ning uudsuse põhjal.

Telefoni tüübid on määratud loeteluna ehk *enumerationina*. Luues uut komparaatorit, võetakse arvesse ka loetelu järjekorda. Telefonitüübid vastavalt tähtsuse järjekorrale (Joonis 11). Kui kliendil on mitu sama tüübi alla kuuluvat numbrit, on oluline järgida numbrite uudsust, ehk kõige suurema järjekorranumbriga (*sequenciga*) number oleks kõige ees. Sellise loogika järgi leitud number kuvataksegi kliendile.



```
public enum PhoneType {  
    MOBILE, PHONE, PHONEABROAD, FAX;  
}
```

Joonis 11. *PhoneType* enum klass

Nii telefoni kui ka emaili väljale on lisatud ka validatsioonid. Telefoninumbri sisestamine on kohustuslik ning tema korral valideeritakse, et välja sisu koosneks vaid numbritest, lubatud on ka suunakoodid ning vastavatel juhtudel on ära määratud kindel numbripikkus. Erinevate sisestusvigade korral kuvatakse kliendile ka vastav veateade. Samuti on emaili sisestamine kohustuslik ning tema korral valideeritakse '@' märgi ning lubatud tähtede ning domeeni olemasolu

## Võrguühenduse loomine

Sander Sepp 39502030023

### Taotluse kontaktandmed

Korrektsete kontaktandmed on vajalikud Teile taotlusega seotud teadete, muudatuste ja pakkumiste edastamiseks.

---

Telefon: *	E-mail: *
<input type="text" value="5344 6989"/>	<input type="text" value="sepp1300@gmail.com"/>

Joonis 12. Kontaktandmete sektsioon

### 4.5.2 Tarbimiskoha sisestus/valik

Tarbimiskoha sektsioon jaguneb erinevate vormide vahel kaheks – ühel juhul peab klient ise sisestama oma tarbimiskoha aadressi, teisel juhul tuleb valida tarbimiskoht vastavast nimistust. Aadressi sisestamisel on kohustuslik täita ka lisaväljad, mis olemasoleval tarbimiskohal on juba määratud ning tulevad teenusest.

Tarbimiskoha sisestamise sektsioon koosneb laias laastus kolmest elemendist – aadressi sisestusest, tarbimiskoha tüübi välikust ning katastritunnuse sisestamisest. Kuna uut liitumist on võimalik teha nii olemasolevale tarbimiskohale, millel on aadress olemas, kui ka aadressi mitteomavale tarbimiskohale, siis sisestusväli pakub kliendile ka

võimalike aadresside nimistu. Väljale tähtede sisestamisel käiakse REGIO teenusest küsimas võimalikke aadresse (Joonis 13). Aadress on kohustuslik valida nimistust, muidu väli ei valideeru ning taotlust pole võimalik esitada.



The screenshot shows a web form titled "Tarbimiskoht (RU)". Below the title is a label "Address: (RU) \*". A text input field contains "Ehitajate tee 5". A dropdown menu is open, listing several address options, each followed by its postal code and county: "Ehitajate tee 5, 12618 Tallinn, Harju maakond", "Ehitajate tee 5, 90401 Uuemõisa alevik, Ridala vald, Lääne maakond", "Ehitajate tee 5, 88101 Tõstamaa alevik, Tõstamaa vald, Pärnu maakond", "Ehitajate tee 5/1, 12618 Tallinn, Harju maakond", "Ehitajate tee 5/10, 12618 Tallinn, Harju maakond", "Ehitajate tee 5/2, 12618 Tallinn, Harju maakond", "Ehitajate tee 5/3, 12618 Tallinn, Harju maakond", "Ehitajate tee 5/4, 12618 Tallinn, Harju maakond", "Ehitajate tee 5/5, 12618 Tallinn, Harju maakond", and "Ehitajate tee 5/6, 12618 Tallinn, Harju maakond". To the right of the dropdown, there is a partially visible label "tsi. (RU)" and another input field with a red "X" and a question mark icon.

Joonis 13. Tarbimiskoha aadressi sisestamise sektsioon

Kuna uut liitumist peab saama teha ka täiesti uuele, veel ilma REGIO aadressita tarbimiskohale, siis on pea võimatu ennustada, kuidas klient kirjutab oma aadressi. Seega loodi eraldi sektsioon "Soovin sisestada aadressi käsitsi". Sinna on loodud kindel väljade struktuur, mis hiljem koodis formuleeritakse ühtse formaadiga aadressiks. Kohustuslik on valida maakond ning sisestada linna või valla nimi. Ülejäänud väljad on kliendile vabatahtlikud, kuna kõik neist väljadest pole aadressis alati määratud. Kõik väljad Joonisel 14 omavad ühtset valideerimistingimust – keelatud on sisestada kirillitsa tähti. Validatsioon loodi seepärast, et Eestis on venekeelsete inimeste arvukus päris suur ning info vahetamise selguse aspektist on aadressi mõistlik saata eesti keeles. Arendamise käigus tekkis küsimus, kas keelata kirillitsa tähestik või lubada vaid eesti keele tähed. Kuna eesti keeles lubatud märke on tunduvalt rohkem kui kirillitsa tähti, osutuski viimaste piiramine kõige sobivamaks.

### Tarbimiskoht

Soovin sisestada aadressi käsitsi

**Maakond: \***  **Linn / Vald: \***

**Küla / Alevik:**  **Tänav:**

**Maja / Talu:**  **Korter:**  **Postiindeks:**

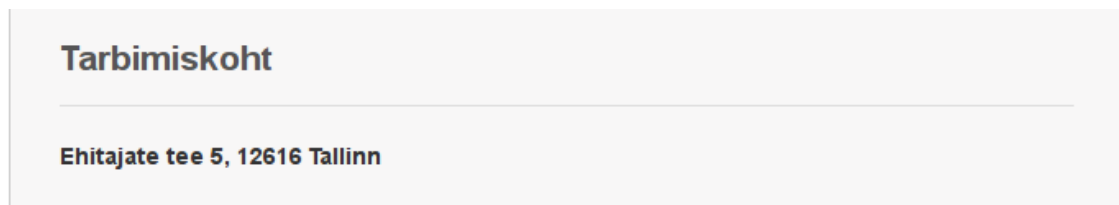
**Tüüp: \***  **Katastritunnus: \***

Joonis 14. Uue aadressi sisestamine

Tüübi valikus on vaid 2 varianti – eramaja ning suvila. Katastritunnuse väli on märksa keerukam. Katastritunnus omab kindlat struktuuri, mis on välja kohahoidjaks ehk *placeholderiks* määratud (XXXXX:XXX:XXXX) (Joonis 14). Klindile välja mugavamaks sisestamiseks loodi väljale reeglite kogum Javascriptis – piisab vaid numbrite sisestamisest, sest vajaminevad koolonid lisatakse ning eemaldatakse automaatselt. Katastritunnuse kopeerimine mujalt antud väljale ei rakenda eelnevalt mainitud Javascripti reegleid. Katastritunnusele lisati Java koodis ka erinevad validatsioonid, kuid nendest lähemalt peatükis 4.8.

Tarbimiskoha valikul on 3 olekut, mille kuva määratakse vastavalt tarbimiskohtade arvule. Kliendi tarbimiskohti käiakse pärimas kliendiinfosüsteemist GetPremisesPrimaryData teenusega. Teenuse väljundis on juba õigel kujul olemas aadress, mida kuvatakse kliendile. Tarbimiskohtade arvu järgi kuvamine jagub gruppidesse järgmiselt: 1, 2-5, 5+ tarbimiskohta. Taoline grupeering loodi pidades silmas kasutusmugavust. 1 tarbimiskoha korral kuvatakse kliendile staatilist aadressi ning minnakse automaatselt edasisteks toiminguteks vajaminevat infot pärima (Joonis 15). 2-5 tarbimiskoha korral genereeritakse kliendile aadressi valik (Joonis 16). Kui kliendil on juba rohkem kui 5 tarbimiskohta, kuvatakse talle tavalist sisestusvälja, millele on külge pandud jQuery UI Autocomplete pistikprogramm. Eelnev *plugin* valiti

võttes arvesse projektis kasutusel olevate tehnoloogiate versioone ning nende ühilduvust (Joonis 17).

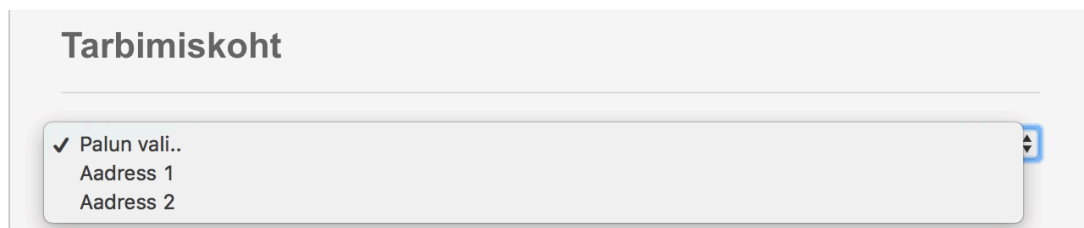


**Tarbimiskoht**

---

**Ehitajate tee 5, 12616 Tallinn**

Joonis 15. Ühe tarbimiskoha kuva

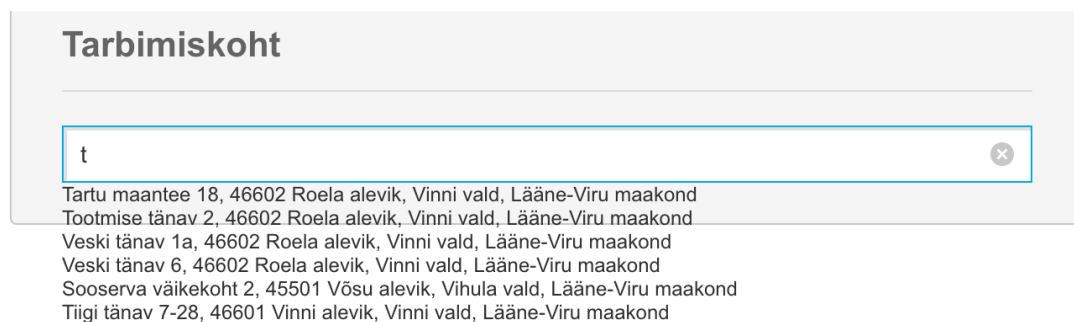


**Tarbimiskoht**

---

✓ Palun vali..  
Address 1  
Address 2

Joonis 16. 2-5 tarbimiskohta kuva



**Tarbimiskoht**

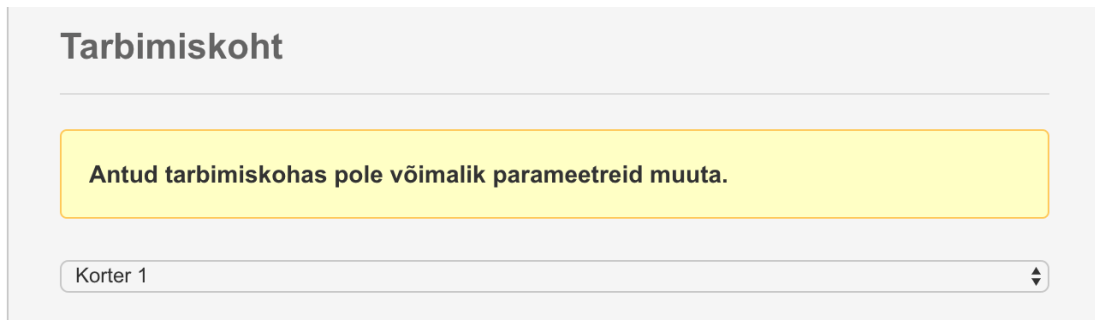
---

t

Tartu maantee 18, 46602 Roela alevik, Vinni vald, Lääne-Viru maakond  
Tootmise tänav 2, 46602 Roela alevik, Vinni vald, Lääne-Viru maakond  
Veski tänav 1a, 46602 Roela alevik, Vinni vald, Lääne-Viru maakond  
Veski tänav 6, 46602 Roela alevik, Vinni vald, Lääne-Viru maakond  
Sooserva väikekoht 2, 45501 Võsu alevik, Vihula vald, Lääne-Viru maakond  
Tiigi tänav 7-28, 46601 Vinni alevik, Vinni vald, Lääne-Viru maakond

Joonis 17. Üle 5 tarbimiskohaga sisestusväli

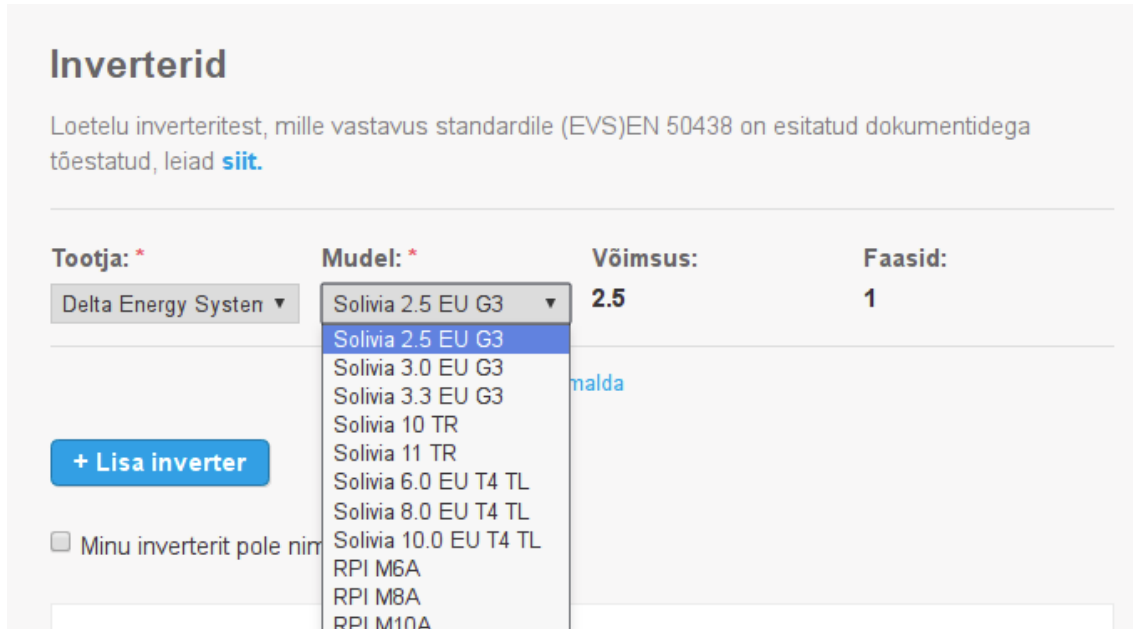
Tarbimiskoha nimekirjas kuvatakse välja kõik teenuselt saadud tarbimiskohad, kuid teatud tüüpi tarbimiskohtadel pole võimalik edasisi tegevusi teha, näiteks korterid (Joonis 18). Nad on sellepärast nimistus välja kuvatud, et kliendil ei tekiks segadust ja ta näeks kõiki enda tarbimiskohti ning ebasobival valikul saaks täiendavat infot. Tarbimiskoha valimisel minnakse erinevatest teenustest küsima infot, mida vastava vormi jaoks on tarvis ning neist on kirjutatud lähemalt järgnevatel peatükkides.



Joonis 18. Infoteade korteri valimisel

### 4.5.3 Mikrotootmisega seotud andmed

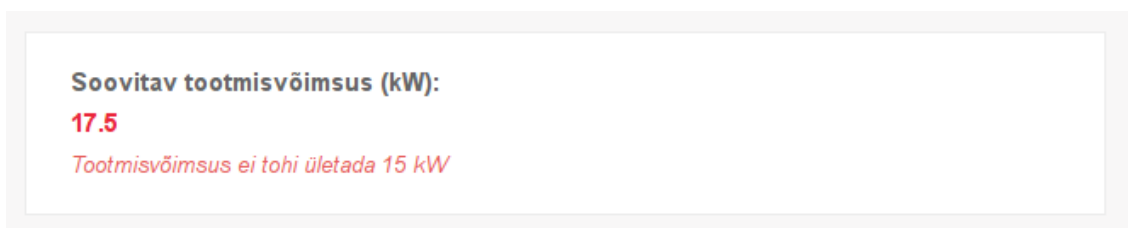
Elektrilevi uute liitumisvormide jaoks loodi teenus `GetAcceptedInvertersForConnection`, mis tagastab inverterite loendi, kus on määratud iga tootja mudelid ning nende põhiinfo – mudeli nimi, tema võimsus ning faaside arv. Tootja valikul pöörduakse *back-endi* poole, mis tagastab konkreetse tootja inverterite mudelid ning väärtustab vajalikud Javascripti muutujad, mille põhjal kuvatakse kliendile vastav info (Joonis 19).



Joonis 19. Mikrotootmisega seotud andmete sektsioon

Kui klient ei leia omale sobivat inverterit pakutud nimistus, on tal võimalik lehel avada uus sektsioon “Minu inverterit pole nimekirjas”. Sellisel juhul tuleb käsitsi sisestada inverteri tootja, mudel, võimsus ning faaside arv.

Inverterite tootmisvõimsus on piiratud 15 kW-ga. Kliendil on õigus lisada mitu inverterit, kuid nende võimsus peab jääma eelpool mainitud piiridesse. Kui valitud pole ühtegi inverterit või inverteerite tootmisvõimsus ületab eelpool mainitud piiri, kuvatakse kliendile vastav veateade (Joonis 17).



Joonis 17. Veateade üle 15kW tootmisvõimsuse korral

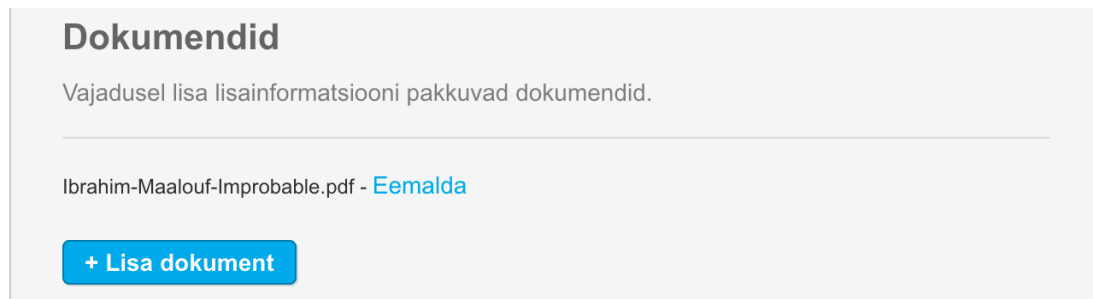
Klient saab invertereid lisada nupust “+ Lisa inverter” ning sellele vajutades genereeritakse dünaamiliselt uus rida, mis kloonib inverteri rea šabloni ning lükkab tolle vastavasse HTML konteinerisse. Igat rida on võimalik ka eraldi kustutada. Võimsuse väljale ning rea eemaldamise külge on Javascripti loodud kuulajad, mis muutuste korral uuendavad soovitava tootmisvõimsuse summat.

Vormi saatmisel antakse teenusesse kaasa kõigi valitud inverterite ID-d, mida liitumiste infosüsteemi poolel osatakse vastu võtta. Kui klient ei leia sobivat inverterit nimistust ning sisestab endale sobiva käsitsi, saadetakse informatsioon teenusesse puhtalt teksti kujul. Taolise olukorra jaoks loodi Liquibase’ga tõlge, kus väärtustatakse teatud parameetrid. Kuna tegemist on kliendi sisestatava väljaga, peab liitumisspetsialist konkreetse inverteri kinnitama ning vajaduse korral kliendiga läbi rääkima, kas too inverter sobib või mitte. Tõlge näeb välja järgmine: “Sooviksin kasutada järgmisi invertereid - {0} {1}, {2} kW, {3} faasi.”. Parameetreid tähistatakse sümboliga “{number}” ning need väärtustatakse Javascriptis enne vormi esitamist.

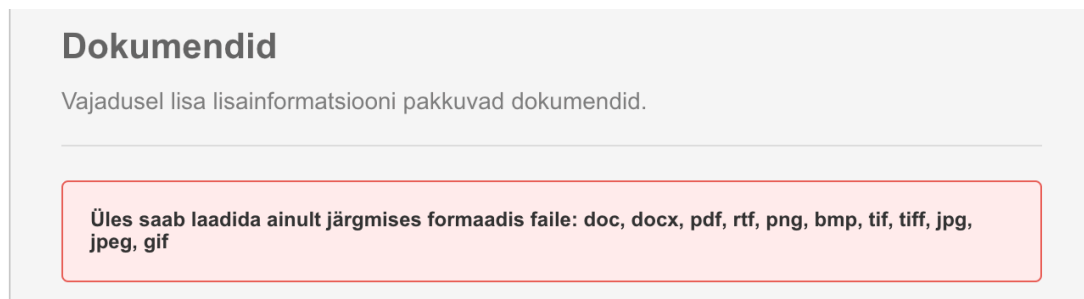
#### 4.5.4 Dokumendid

Failide lisamise sektsioon on vormidel selleks, et klient saaks saata olulist lisainfot oma tarbimiskoha kohta (Joonis 20). Failide lisamisel saab tekkida kahte sorti vigasid – vale laiendiga või liiga suure faili lisamine. Vormide porteletidele, millel on dokumentide

lisamise sektsioon, on loodud konfiguratsiooni muutuja, kus saab ära määrata lubatud faililaiendid. Lubatud laiendid loetakse sisse enne vaate genereerimist abstraktses kontrollis. Vales formaadis faili sisestamisel kuvatakse kliendile veateade, kus on välja toodud kõik portleti konfiguratsioonis mainitud faililaiendid (Joonis 21).

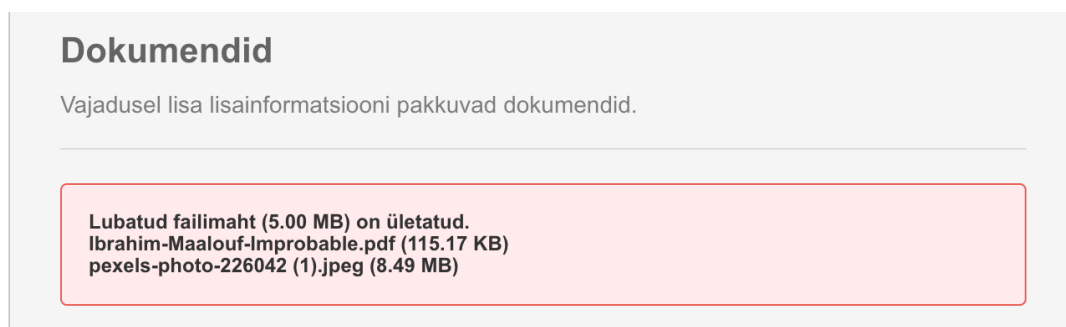


Joonis 20. Dokumentide sektsioon



Joonis 21. Veateade vale laiendiga faili sisestamisel

Sisestavate failide arv ei ole piiratud, küll aga on seatud failimahu ülempiir, milleks on üle kogu süsteemi 5MB. Juhul kui failide kogumaht ületab piiri, kuvatakse kliendile järgnev veateade, kus on välja toodud kõik failid koos nende suurustega (Joonis 22). Seeläbi saab klient teha kõige optimaalsema valiku.

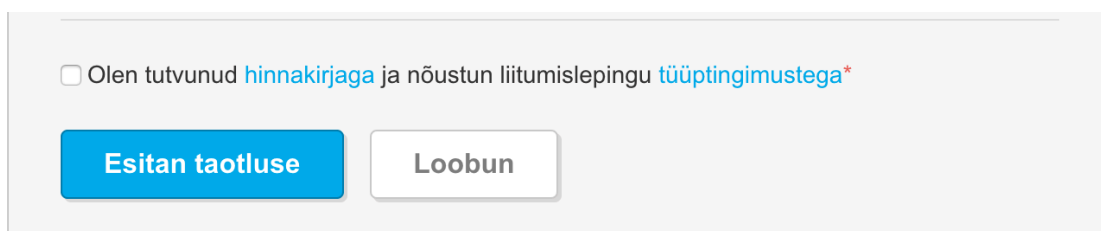


Joonis 22. Mahu ületamisel kuvatav veateade

#### 4.5.5 Tingimustega nõustumine

Tingimustega nõustumise sektsioon on sarnaselt kontaktandmetega esindatud igas vormis. Taotluse esitamiseks peab klient nõustuma erinevate tingimustega ning tegema linnukese *checkboxis* (Joonis 23), vastasel korral vorm ei valideeru ning taotlust pole võimalik edastada.

Nupp “Esitan taotluse” edastab vormi *back-end*-i, kus valmistatakse ette kogu vormi andmekomplekt, mis saadetakse teenusesse. Igale portletile on loodud konfiguratsioonimuutuja, mis määrab ära asukoha, kuhu viiakse vajutades nuppu “Loobun”.

The image shows a user interface element for consent. It consists of a light gray rectangular box. At the top, there is a text line: "Olen tutvunud [hinnakirjaga](#) ja nõustun liitumislepingu [tüüptingimustega](#)\*". The text is in black, with the links in blue. Below this text, there are two buttons. The first button is blue with white text that says "Esitan taotluse". The second button is white with a gray border and black text that says "Loobun".

Joonis 23. Tingimustega nõustumise sektsioon

#### 4.6 Uus liitumine madalpingel

Uue liitumise vorm koosneb järgnevatest sektsioonidest:

- 1) Kontaktandmed
- 2) Aadress
- 3) Liitumispunkti parameetrid
- 4) Liitumispunkti asukoht kaardil
- 5) Mikrotootmisega seotud andmed
- 6) Orienteeruv hind
- 7) Dokumentide lisamine
- 8) Tingimustega nõustumine



#### 4.6.1 Liitumispunkti parameetrid

Sektsioon liitumispunkti parameetrid koosneb kahest osast: faasi valik ja peakaitsme suuruse valik. Liitumistaotlust saab teha nii 1 kui ka 3-faasilisele ühendusele. Peakaitsete jaoks on 2 loendit, sest erinevate faaside korral on võimalik valida erinevaid amprite suurusi.

1-faasilise ühenduse korral on võimalik valida järgnevaid peakaitsme väärtusi: 1 A, 2 A, 3 A, 4 A, 6 A, 10 A, 16 A, 20 A, 25 A. 3-faasilise korral võimalikud väärtused järgmised: 1 A, 2 A, 3A, 4A, 6A, 10A, 16A, 20 A, 25A, 32 A, 40 A, 50 A, 60 A, 63 A, 80 A, 100 A, 125 A, 160 A, 180 A, 200 A, 225 A, 250, 280 A, 315 A, 350 A, 400 A, 500 A, 560 A, 600 A, 630, 700 A, 750A, 800 A, 900 A, 1000 A, 1125 A, 1250 A, 1280 A, 1440 A, 1600 A, 1800 A, 2000 A, 2250 A, 2500 A. Eelpool nimetatud väärtusi vahetatakse peakaitsme valikus faaside muutmisel. Amprite väärtusest sõltub ka ligikaudne liitumise hind – mida rohkem ampreid, seda suurem hind.



The screenshot shows a web form titled "Liitumispunkti parameetrid". It contains two dropdown menus. The first is labeled "1 või 3 faasi:" and has "3 faasi" selected. The second is labeled "Peakaitsme suurus" and has "4 A" selected. A question mark icon is located to the right of the first dropdown menu.

Joonis 24. Liitumispunkti parameetrid

#### 4.6.2 Liitumispunkti asukoht

Avades uue liitumise vormi, pole kliendile veel kuvatud sektsiooni "Liitumispunkti asukoht". Mainitud sektsioon genereeritakse valmis vaid juhul, kui klient on sisestanud realselt temale kuuluva katastritunnuse. Katastritunnust minnakse valideerima vajutades nuppu "Valideerin tarbimiskoha" (Joonis 25). Esmalt pöörduakse läbi teenuse Registrate ja Infosüsteemide keskuse poole, kus hoitakse infot katastrite ning nende omanike kohta. Kui sisestatud kataster ei kuulu isikule, kelle nimel taotlust vormistatakse, kuvatakse katastri välja alla vastav veateade (Joonis 25). Juhul, kui kataster kuulub vastavale isikule, küsitakse teenusest GetCadasterData maa-ala piirjooni ning genereeritakse liitumispunkti asukoha sektsioon, kus on muust kaardist eraldatud katastri alla kuuluv piirkond.

**Tüüp: \***

Eramaja ▼

**Katastritunnus: \***

19801:001:0748 ?

*See kinnistu ei kuulu isikule XXX, liitumist saab taotleda vaid kinnistu omanik või tema poolt volitatud.*

### Liitumispunkti parameetrid

---

**1 või 3 faasi:**

3 faasi ▼ ?

**Peakaitse suurus**

6 A ▼

Valideerin tarbimiskoha

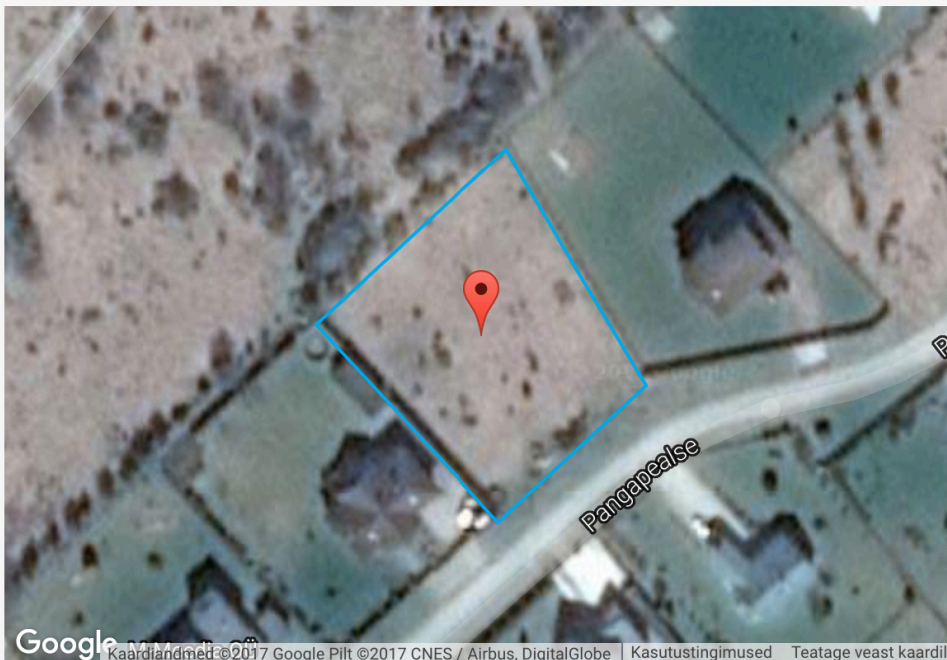
Joonis 25. Tarbimiskoha valideerimine

Vaikimisi asetatakse punane marker kaardi keskele (Joonis 26). Kliendil on võimalik markerit kaardi raames ümber paigutada. Vormi laadimisel pannakse andmemudelisse kaasa parameeter KMP ehk kindlaksmääratud piirkond. Too muutuja on konfigureeritav Liferay-st ning määrab ära maksimaalse kauguse liitumispunktist, mille korral pole vajadust suuremateks võrgutöödeks.

Markeri kaugust liitumispunktist minnakse kontrollima vajutades nuppu “Kontrollin liitumispunkti asukohta”. Kauguse kontrollimisel pöördutakse teenuse FindCustomerPointElectricalParameters poole, kuhu sisendina lähevad sisse markeri koordinaadid ning väljundisse tuleb konkreetne kaugus liitumispunktist täisarvuna. Vastavalt teenuse vastusele kuvatakse kliendile teadet, kas valitud koht asub KMP-s või mitte. Kõigest eelnevast sõltub ka hinnaarvutus – kui liitumiskoht asub KMP-s, siis kuvatakse kliendile liitumise ligikaudne hind, vastasel korral näidatakse teksti “Hind kujuneb pakkumise käigus tegelike kulude alusel”. Lisainfo välja on taotluse esitajal võimalik kirjutada olulist maa-ala või liitumiskoha kohta.

## Liitumispunkti asukoht

Märkige kaardil tarbimiskoha liitumispunkt ehk kuhu soovite kilbi paigaldada. Liitumispunkt peaks olema krundi piiril. Suurema kinnistu korral maja läheduses õuema piiril..



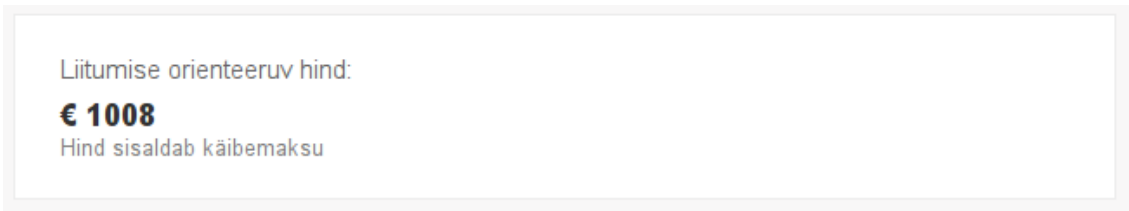
Kontrollin liitumispunkti asukohta

Lisainfo:

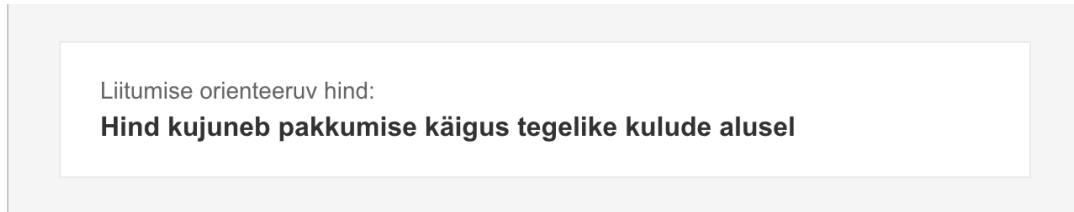
Joonis 26. Liitumispunkti asukoht

### 4.6.3 Liitumise orienteeruv hind

Hinnaarvutus on Elektrilevi kõikide vormide juures uus funktsionaalsus. Uue liitumise vormil sõltub hind päris mitmest asjaolust – peakaitsme suuruselt ja liitumiskoha paiknemisest KMP-s. Sektsioonis on kas hinnakuva (Joonis 27) või info, et selliste parameetrite juures pole võimalik orienteeruvat hinda määrata (Joonis 28).



Joonis 27. Orienteeruva hinna kuva



Joonis 28. Tegelike kulude info kuva

Kliendile ei kuvata orienteeruvat hinda, kui soovitakse liituda mikrotootjana. Taolisel juhul on keeruline hinnata tööde mahtu, kuna iga liitumise tingimused ning nõuded on erinevad. Teine olukord, mil hinda ei arvutata, on sellisel juhul, kui liitumispunkti ning markeri ehk reaalse liitumiskoha vaheline kaugus ületab maksimaalset vahemaad ehk ei asu kindlaks määratud piirkonnas (KMP), mis on seadistatav Liferay-s.

Orienteeruvat hinda kuvatakse juhul, kui klient ei ole avaldanud soovi liituda mikrotootjana ning valitud liitumiskoht asub KMP-s. Hinna suurus sõltub peakaitsme väärtusest, mida on võimalik muuta vormil veidi üleval pool. Amprite muutmisel pöörduetakse teenuse GetConnectionProcessPrices poole, mille väljundisse tuleb kahte tüüpi hindasid – ampritasu ning menetlustasu. Amprite, ampritasu ning menetlustasuga tehakse vastavad arvutused, mille tulemus kuvataksegi kliendile.

## 4.7 Tarbimistingimuste muutmine

Tarbimistingimuste muutmise vorm koosneb järgnevatest sektsioonidest:

- 1) Kontaktandmed
- 2) Tarbimiskoha valik
- 3) Peakaitsme toode

- 4) Liitumispunkti asukoha toode
- 5) Võrguühenduse faaside toode
- 6) Orienteeruv hind
- 7) Tingimustega nõustumine

Tarbimistingimusi on võimalik muuta tarbimiskohtades, mis ei ole alamtarbimiskohad (korterid). Vormi põhiosadeks on tooted, mida klient saab valida – peakaitsme, liitumispunkti asukoha ning võrguühenduse faaside toode. Eelnevalt nimetatud esimesed 2 toodet on kuvatud kõikidele tarbimiskohtadele.

#### **4.7.1 Tooted**

Tarbimiskohta valides küsitakse teenusest `GetServicePointPrimaryData` tema primaarandmed, millest oluline salvestatakse vaate andmemudelisse. Kui valitud tarbimiskohal on 1-faasiline ühendus, ainult sel juhul kuvatakse välja võrguühenduse faaside toode. Peakaitsme suuruse tootel on all väike selgitustekst, kus on ära määratud maksimaalne peakaitsme suurus, mille arvutus tehakse Javascriptis enne toodete sektsiooni genereerimist. Näiteks kui kliendi suurim võimalik peakaitse on 560 amprit, siis mainitud väärtus on suurim ka peakaitsme valikus.

### Tarbimiskoha parameetrite muutmine

Soovin muuta peakaitsme suurust

**Praegune peakaitse:** **16A**      **Soovitud peakaitse:**

Saame pakkuda Teile peakaitset suurusega kuni **40A** kaasa arvatud ilma suuremate ümberehitustöödeta. **Kui soovite sellest suuremat peakaitset on vajalikud ümberehitustööd.**

Soovin muuta faaside arvu

**Soovin üle minna 3-faasilise võrguühendusele**

Soovin muuta liitumispunkti asukohta

**Praegune asukoht:**  
**Õhuliini sisestusvisangu isolaatoritel hoone seinal**

Joonis 29. Peakaitsme suurendamise, liitumispunkti asukohta muutmise ja faaside suurendamise tooted

#### 4.7.2 Hinnaarvutus

Võrreldes uue liitumise vormiga on tarbimistingimuste muutmise hinnaarvutuse kuvamine märksa keerulisem. Igal tootel on oma tootekood. Pealtnäha on valikus 3, kuid süsteemi jaoks eksisteerib 6 erinevat toodet. Järgnevalt on välja toodud mainitud tooted:

- 1) PK suurendamine KMP-s
- 2) PK suurendamine väljaspool KMP-d
- 3) PK vähendamine, kui olemasolev PK on kuni 63A kaasa arvatud
- 4) PK vähendamine, kui olemasolev PK on suurem kui 63A
- 5) Üleminek 1-faasiliselt 3-faasilisele ühendusele

## 6) Liitumispunkti asukohta muutmine

Vormi saatmisel tuleb väärtustada põhitoode ning valitud lisatooded. Eelpool väljatoodud nimekirjas on tooted juba olulisuse järjekorras. Näiteks kui olemasolevat PK-d soovitakse suurendada ning liitumispunkti asukohta muuta, siis põhitoodeks on toode 1 ning lisatooteks toode 6. Seega toote valikul pööratakse *back-end*-i poole, kus määratakse ära põhitoode ning lisatooded, mille alusel kuvatakse orienteeruva hinna sektsiooni.

Orienteeruvat hinda kuvatakse, kui klient on valinud toote 1 või 3 ja valitud ei ole toode 5 ja/või 6 ning maksumuse suurus sõltub peakaitsme tootes valitud amprite arvust. Hinda ei kuvata, kui on valitud tooted 2, 4, 5 ja/või 6. Kuna toodete määramisega kaasneb üsna palju loogilisi tegevusi, otsustati funktsionaalsus realiseerida ainult *back-end*-is, kuhu tehti peale ka ühiktestid.

## 4.8 Elektrivõrgu ümberehitus

Elektrivõrgu ümberehituse vorm koosneb järgnevatest sektsioonidest:

- 1) Kontaktandmed
- 2) Aadress
- 3) Dokumentide lisamine
- 4) Soovi kirjeldus (lisainfo)
- 5) Tingimustega nõustumine

Tegemist on Elektrilevi liitumisvormide kõige lihtsama vormiga. Erinevalt teistest vormidest ei oma ta ühtegi unikaalset sektsiooni. Elektrivõrgu ümberehitust saab tellida nii uuele kui ka olemasolevale tarbimiskohale. Seega on tarbimiskoha valiku asemel hoopis aadressi sisestamise sektsioon – kliendil on võimalik sisestada aadressi, mis on olemas REGIO andmebaasis või täiesti uus. Kuna kood ehitati ülesse väga modulaarselt, oli võimalik vajalikke sektsioone hõlpsasti kaasata.

## 4.9 Mikrotootja liitumine

Elektrivõrgu ümberehituse vorm koosneb järgnevatest sektsioonidest:

- 1) Kontaktandmed
- 2) Tarbimiskoha valik
- 3) Olemasoleva võrguühenduse andmed
- 4) Mikrotootmisega seotud andmed
- 5) Inverterite valik
- 6) Tingimustega nõustumine

### 4.9.1 Võrguühenduse andmed

Tarbimiskoha valikul küsitakse teenuselt `GetServicePointPrimaryData` teenuspunkti primaarandmed. Võrgulepingu numbrit, mõõtepunkti EIC koodi, arvesti numbrit, peakaitsme suurust ja faaside arvu kuvatakse kliendile ka välja (Joonis 30). Samuti on järgnevad tegevused seotud eelnevate andmetega.

Võrguühenduse andmed		
Võrgulepingu number: 5552123490	Mõõtepunkti EIC kood: 38ZEE-0012345-L	Arvesti number: 191231611
Peakaitsme suurus: 20A	Faaside arv: 3	

Joonis 30. Võrguühenduse andmed

### 4.9.2 Mikrotootmisega seotud andmed

Too sektsioon on ka uue liitumise vormil, kuid mikrotootja liitumisel lisandub üks alamsektsioon, mis koosneb kahest väljast – praegune tootmisvõimsus ning inverterite arv. See info saadakse samuti teenusest `GetServicePointPrimaryData` nagu võrguühenduse andmete korralgi. Enne Elektrilevi liitumisvormide arendust praeguse tootmisvõimsuse ning inverterite arvu välju teenuses ei olnud, seega tuli need sinna lisada (Joonis 31).



## Mikrotootmisega seotud andmed

Mikrotootja on elektrienergia tootja, kes toodab elektrit väikeses koguses taastuvatest allikatest, peamiselt oma majapidamise või ettevõtte tarbeks. Mikrotootmiseseadmeks võib olla **ühefaasiline** tootmiseseade, mille nimivõimsus on kuni **3,68 kW**, või **kolmeefaasiline** tootmiseseade, mille nimivõimsus on kuni **15 kW**.

**Praegune tootmisvõimsus:**  
**10kW**

**Inverterite arv:**  
**5**

Joonis 31. Tarbimiskoha mikrotootmise info

Mikrotootmisega seotud andmete juures saab klient endale lisada ka uusi inverterid, mis on lahti kirjutatud alampunktis 4.7.3. Maksimaalne tootmisvõimsus on mikrotootmise korral endiselt 15kW, seega kui klient juba eelnevalt omab inverterid, arvestatakse uute lisamisel ka vanu. Mikrotootmise taotlust saab teha ainult olemasolevale tarbimiskohale, mistõttu on peakaitse suurus alati määratud ning selle pealt on võimalik välja arvutada tarbimiskoha maksimaalne tootmisvõimsus. Kui uute inverterite lisamisel ületatakse eelpool mainitud piir, kuvatakse kliendile uus sektsioon, kus on võimalik suurendada hetkel olemasolevat peakaitset. Invertereid on nii 1 kui ka 3-faasilisi. Seega, kui kasutajal on 1-faasiline tarbimiskoht ning ta valib inverteerite hulgast 3 faasilise inverteri, pakutakse kliendile võimalust suurendada oma faaside arvu 3 peale (Joonis 29).

Soovin muuta peakaitsme suurust  

<b>Praegune peakaitse:</b> 16A	<b>Soovitud peakaitse:</b> 20 A ▼
-----------------------------------	--------------------------------------

Saame pakkuda Teile peakaitset suurusega kuni **40A** kaasa arvatud ilma suuremate ümberehitustöödeta. **Kui soovite sellest suuremat peakaitset on vajalikud ümberehitustööd.**

Soovin muuta faaside arvu  

<b>Praegune võrguühendus:</b> 1 faasiline	<input checked="" type="checkbox"/> Soovin üle minna 3-faasilise võrguühendusele
--	--

Joonis 32. Peakaitsme suurendamine ja faaside muutmine

#### 4.10 Tehnilised tingimused

Elektrivõrgu ümberehituse vorm koosneb järgnevatest sektsioonidest:

- 1) Kontaktandmed
- 2) Aadress
- 3) Tasuta ja tavalised tingimused
- 4) Dokumentide lisamine
- 5) Tingimustega nõustumine

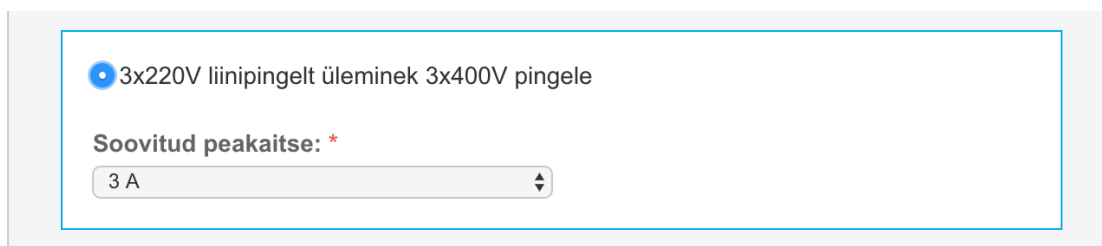
Antud vormi unikaalseteks sektsioonideks on tasuta ning tasulised tingimused. Vormil saab valida vaid ühe toodetest, pole vahet, kas tasuta või tasulise tingimuse. Pea-aegu iga tingimuse valimisel avaneb kliendile sektsioon, kus on võimalik täpsemalt tellitavat tingimust seadistada. Üle mitme liitumisvormi on levinud peakaitsme ning faaside valikud, mida saab ilusti ka antud vormi raames kasutada. Juurde tuli luua keskpinge võimsuse sisestamise kastikene.

#### 4.10.1 Tasuta tingimused

Tasuta tingimusi on 3, mis on järgmised:

- 1) 3x220V liinipingelt üleminek 3x400V pingele
- 2) Liitumine keskpingel
- 3) Tehnilised tingimused hoone elektriprojekti koostamiseks

Tingimus number 1 korral avaneb sektsioon, kus on kuvatud vaid peakaitsme valik (Joonis 33). Kuna tegemist on ainult 3-faasilise võrguühenduse tingimusega, pole kliendil võimalust faase eraldi valida ning valitavad amprite väärtused vastavad 3-faasilise ühendusele. Tingimus 2 puhul avaneb kliendile keskpinge võimsuse sisestuskast (Joonis 34). Erinevalt amprite valikust pole võimsusele võimalik kindlaid väärtuseid ette anda, seega saab klient sisestada mistahes numbri. Tingimust number 3 on võimalik taotleda nii 1 kui ka 3-faasilisele võrguühendusele. Seega peab lisaks peakaitsme valikule olema võimalik määrata ühenduse faaside arvu. Vastavalt viimasele, väärtustatakse ka vastavad peakaitsme väärtused (Joonis 35).

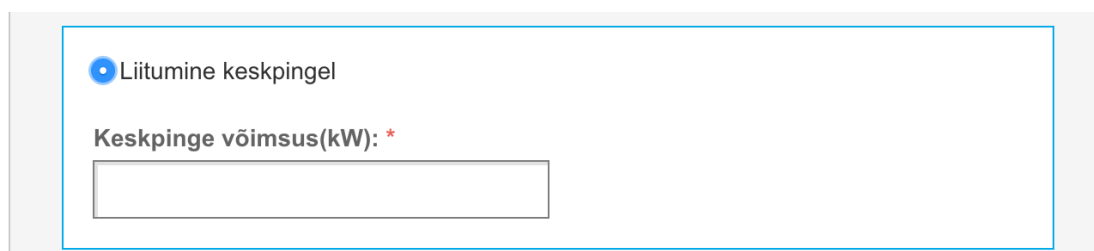


3x220V liinipingelt üleminek 3x400V pingele

Soovitud peakaitse: \*

3 A

Joonis 33. 3x220V liinipingelt üleminek 3x400V pingele



Liitumine keskpingel

Keskpinge võimsus(kW): \*

Joonis 34. Liitumine keskpingel

Joonis 35. Tehnilised tingimused hoone elektriprojekti koostamiseks

#### 4.10.2 Tasulised tingimused

Tasulisi tingimusi on kokku 4:

- 1) Liinimastide ühiskasutus
- 2) Detailplaneering
- 3) Ajutine võrguühendus
- 4) Mitteelektriprojekti koostamine

Tingimuse 1 ja 4 korral pole kliendil vaja midagi täpsemat sisestada (Joonised 33 ja 34). Nii tingimuse 2 kui ka 3 korral on võimalik määrata kas võimsust või peakaitsme suurust (Joonised 35-38). Võimsuse sisestamisel on kasutatud sama komponenti, mis keskpinge võimsuse korralgi. Amprite valikul on valimiks 3-faasilise võrguühenduse väärtused. Detailplaneeringu juures on kliendile oluline kuvada teatud infot – võimsuse sisestamisel vajalikku amprite arvu ning vajalikku peakaitsme suurust, peakaitsme valikul vajalikku võimsust. Info kuvamine on realiseeritud läbi parametriseeritud tõlgetele, kuhu lähevad sisenditeks arvutused läbinud tulemused.

Joonis 36. Liinimastide ühiskasutus

Joonis 37. Mitteelektriprojekti koostamine

Detailplaneering  
 Võimsus (kW) või  Peakaitse (A)

**Võimsus (kW): \***

12

12 kW juures on vajalik amprite arv 18.12 A.  
Vajalik peakaitsme suurus on 20 A.

Joonis 38. Detailplaneeringu võimsuse sektsioon

Detailplaneering  
 Võimsus (kW) või  Peakaitse (A)

**Soovitud peakaitse: \***

40 A

40 A peakaitsme suuruse juures on vajalik võimsus 26.40 kW.

Joonis 39. Detailplaneeringu peakaitsme sektsioon

Ajutine võrguühendus  
 Võimsus (kW) või  Peakaitse (A)

**Võimsus (kW): \***

12

Soovin hinnapakkumist ajutise võrguühenduse kilbi rendiks koos paigalduse ja hilisema lahtiühendamisega

Joonis 40. Ajutise võrguühenduse võimsuse sektsioon

Ajutine võrguühendus  
 Võimsus (kW) või  Peakaitse (A)

**Soovitud peakaitse: \***

32 A

Soovin hinnapakkumist ajutise võrguühenduse kilbi rendiks koos paigalduse ja hilisema lahtiühendamisega

Joonis 41. Ajutise võrguühenduse peakaitsme sektsioon

## 4.11 Testimine

Ühiktestid on loodud funktsionaalustele, kus tuleb vormile kuvada mingeid kindlaid väärtusi või välja lugeda kindlad toodete kombinatsioonid. Kasutajapoolset testimist

viivad läbi Eesti Energia testijad, kes vaatavad üldist vormi funktsioneerimist – valideerivad kuvatud andmete õigust ning nende jõudmist liitumiste infosüsteemi. Leitud vead dokumenteeritakse Eesti Energia Confluence'i keskkonda.

Iga vormi alguses olevas kontaktandmete sektsioonis on oluline kuvada kliendile õiget kontakttelefoni numbrit. Telefonitüüpide olulisuse järjestus on lahti kirjutatud peatükis 4.6.1 ehk mobiil, lauatelefon, välismaa telefon ning faks. Sama tüübi korral on oluline vaadata suurima *sequence* numbriga telefoni (Joonis 42). Seega kirjutati ühiktestid kõikidele võimalikele olukordadele ning seeläbi oli hea kood ülesse ehitada nii, et ei oleks vaja kasutada *mockimist*.

```
@Test
public void getPhoneNumberWithLargestSequenceNumberShouldReturnMobileWithHighestSequenceNumber() {
    PhoneNumber number1 = getPhoneNumber(PhoneType.MOBILE, sequenceNumber: 1);
    PhoneNumber number2 = getPhoneNumber(PhoneType.MOBILE, sequenceNumber: 2);
    PhoneNumber number3 = getPhoneNumber(PhoneType.PHONE, sequenceNumber: 3);
    PhoneNumber number4 = getPhoneNumber(PhoneType.PHONEABROAD, sequenceNumber: 4);
    assertEquals(number2, contactInfoHelper.getPhoneNumberWithLargestSequenceNumber(Arrays.asList(number1, number2, number3, number4)));
}
```

Joonis 42. Näide telefoninumbri testist

Üheks olulisemateks osutused käsitsi sisestavate aadresside testid, sest erinevaid sisestusvõimalusi on palju ning tähtis on aadress formaatida sarnaselt REGIO aadresside struktuurile. Liitumiste infosüsteemi jaoks on oluline kokku panna sisestatu kahes formaadis – üks väärtustatakse *address* ja teine *location* väljale. Aadressi väärtustamise puhul on 2 varianti, kas väikeasula on väärtustatud või mitte. Kui viimast pole sisestatud rakendub järgmine loogika: linn, tänav, maja ja olemasolul korter. Vastasel juhul väikeasula, tänav, maja ja olemasolul korter. *Locationi* väärtustamisel on elementide eraldajateks komade asemel tühikud. Kui väikeasula on väärtustatud, genereeritakse *location* järgmiselt: “väikeasula linn maakond”, vastasel korral “linn maakond”.

Vead, mille avastasid Eesti Energia testijad, dokumenteeriti Confluence'i keskkonda. Sealne jaotus laias laastus kaheks – vead, mis olid seotud vormide täitmise ja arvutustega ning vead, mis tekkisid andmevahetusest liitumiste infosüsteemi ja Elektrilevi iseteeninduse vahel. Testimise tulemusena tuli muuta paari teenust ning vormil kuvatavat infot.

## 5 Kokkuvõte

Töö põhieesmärgiks oli täiendada Elektrilevi iseteeninduskeskkonda luues juurde liitumisvorme ning uuendades vanu. Mikrotootja liitumist ja tehniliste tingimuste muutmist oli võimalik senini sõlmida vaid paberil, veebis puudusid vastavad vormid. Uue liitumise, tarbimistingimuste ja elektrivõrgu ümberehituse vormidele loodi juurde funktsionaalsust. Projekt on lõplikult valmis ning testitud erinevates testkeskkondades. Arenduste *live*’i paigaldamiseks on ette nähtud igakuised *release*’id ning antud lõputöös tehtud vormid lähevad juuni kuus toodangusse. Seega reaalselt ei ole klientidel veel võimalik tulemust kasutada. Tööst on mõjutatud päris suur hulk inimesi – liitumistaotlusi esitatakse aastas umbes 12000 tükki.

Paralleelselt Elektrilevi iseteeninguse vormidega arendati välja liitumiste infosüsteem, mis kasutab Camunda BPMN-i. Seal on võimalik paremini hallata ning automatiseerida äriprotsesse. Näiteks taotluse esitamisel on võimalik esitatud käidupiirkonna põhjal määrata töö konkreetse piirkonna liitumisspetsialistile. Seal on selge ülevaade, milliseid töid keegi teeb ning mis iseteenindusest tulnud taotlustega on tegu. Lepingu sõlmimiseks ei piisa vaid ühest sammust. Tegemist on pikema protsessiga – taotluse esitamine, pakkumise tegemine, lepingu allkirjastamine ja töö realiseerimine. Kõike seda on võimalik liitumiste infosüsteemis teha tänu iseteeninduse uutele vormidele.

Arenduse algfaasis oli enamik informatsiooni äriloogika kohta juba olemas ning prototüübid olid valmimises. Kogu protsess oli väga agiilne ning arvukad koosolekud Elektrilevi projektijuhtidega täiendasid lõplikku tulemust veelgi. Siin-seal tulid välja puudused kasutajakogemuse kui ka reaalse teenustevahelise suhtluse aspektidest. Projekti edukuse tagaski just erinevate inimeste aktiivsus ning professionaalsus. Kuna äriloogika maht oli küllalt suur, oldi alati abivalmis kõike seletama ning vastused erinevatele keerulistele küsimustele leiti võrdlemisi kiirelt.

Antud bakalaurusetöö oli heaks katsumuseks äriinfotehnoloogia tundengile. Seda just seepärast, et eriala on pigem suunatud ärivaldkonnale kui reaalsele programmeerimisele, kuid eriti hinnatakse oskust neid kahte kombineerida.

Lõppkokkuvõttes oli tulemus igati positiivne ning karastas nii mõnekski olukorraks, mis tuleviku vaates osutub kindlasti väga kasulikuks.



## Kasutatud kirjandus

1. Apache Ant – Welcome. [WWW] <http://ant.apache.org/> (29.04.2017)
2. Apache Axis2/Java – Next Generation Web Services. [WWW] <http://axis.apache.org/axis2/java/core/> (27.04.2017)
3. Beust, C. (2015). TestNG. [WWW] <http://testng.org/doc/> (28.04.2017)
4. Dugas, C. (2016). jQuery.validationEngine v2.6.2. [WWW] <https://github.com/posabsolute/jquery-Validation-Engine> (30.04.2017)
5. Metsis, M. (2013). Arhitektuur. [WWW] Asutusesisene dokument. (27.04.2017)
6. McGarr, M. (2010). Manage Database Changes with Liquibase. [WWW] <https://earlyandoften.wordpress.com/2010/06/28/intro-to-liquibase/> (09.05.2017)
7. Ozawa, H. (2011). About Portlets. [WWW] <https://web.liferay.com/community/wiki/-/wiki/Main/About+Portlets> (26.04.2017)
8. Sanderson, S. (2009). Writing Great Unit Tests: Best and Worst Practises. [WWW] <http://blog.stevensanderson.com/2009/08/24/writing-great-unit-tests-best-and-worst-practises/> (29.04.2017)
9. Sezov, R. (2016). Liferay's database. [WWW] [https://dev.liferay.com/discover/deployment/-/knowledge\\_base/6-2/liferays-database](https://dev.liferay.com/discover/deployment/-/knowledge_base/6-2/liferays-database) (26.04.2017)
10. Sezov, R. (2016). What is Liferay? [WWW] [https://dev.liferay.com/discover/portal/-/knowledge\\_base/6-2/what-is-liferay](https://dev.liferay.com/discover/portal/-/knowledge_base/6-2/what-is-liferay) (26.04.2017)
11. Spring MVC Framework. [WWW] [https://www.tutorialspoint.com/spring/spring\\_web\\_mvc\\_framework.htm](https://www.tutorialspoint.com/spring/spring_web_mvc_framework.htm) (27.04.2017)
12. The DispatcherPortlet. [WWW] <http://docs.spring.io/spring/docs/current/spring-framework-reference/html/portlet.html#portlet-dispatcher> (30.04.2017)
13. Tint, M. (2014). Home. [WWW] Asutusesisene dokument. (27.04.2017)