

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Martin Siitam 206097IADB

# **Varahalduse ja aruandluse lahendus päikeseparkide pakkujale ja selle klientidele**

Bakalaureusetöö

Juhendaja: Meelis Antoi  
Magistrikraad

Kaasjuhendaja: Taavi Sarnet  
Magistrikraad

Tallinn 2024

## **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: Martin Siitam

03.01.2024

## **Annotatsioon**

Käesoleva bakalaureusetöö eesmärgiks oli luua töötav varahalduse ja aruandluse lahendus päikeseparkide haldajale, paigaldajale ja selle klientidele.

Arendusprotsessi käigus loodi veebirakendus, mis võimaldab varahalduritel, operaatoritel, projektijuhtidel ning era- ja äriklientidel saada reaalajas ning ajaloolist informatsiooni päikeseparkide tootlikkuse ning alarmide kohta. Eelmainitud informatsiooni põhjal saab terviklahenduse raames päikesepargi terviklahenduse pakkuja teha õigeid otsuseid päikesepargi efektiivse töö tagamisel nii finantsilisel kui ka tehnilisel tasemel.

Lõputööna teostatava projekti raames lõi lõputöö autor ka era- ja äriklientide jaoks raporteerimislahenduse, mis edastab klientidele informatsiooni kuuraportina.

Lahenduse loomise protsess koosnes mitmest osast, mis käsitlesid olemasolevate lahenduste analüüsi, erinevate kasutajagruppide sisendi põhjal skoobi defineerimist, tehnoloogia valiku põhjendamisest ning erinevate osapoolte testimise tulemusena saadud tagasiside põhjal muudatuste elluviimisest.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 53 leheküljel, 8 peatükki, 2 joonist.

## **Abstract**

The aim of the thesis was to create a working asset management and reporting solution for solar asset manager, installer and its clients.

During the development process a web application was created that allows asset managers, operators, project managers, private and business customers to receive live and historical production and alarm data from the solar parks. The aforementioned information will allow the solar park provider to make informed decisions to ensure efficient production on both financial and technical level.

As a part of the project, the author of the thesis will also create a reporting solution that will send the private and business customers an overview of the past months production.

The process for creating the asset management and reporting solution consisted of multiple parts. These topics covered the analysis of existing solutions, mapping the needs of different user groups and using that to define the scope of the project, the process of selecting the technology to implement the solution and finally the process of optimizing and changing the application based on the feedback from testing by different end users,

The thesis is in Estonian and contains 53 pages of text, 8 chapters, 2 figures.

## Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> - rakendusliides
<i>backend</i>	Teenusepoolne keskkond
CSS	<i>Cascading Style Sheets</i> - veebilehtede küljendamisel kasutatav märgistuskeel
CSV	<i>Comma-separated values</i> - komaeraldusega väärtused
<i>frontend</i>	Kasutajaliides, kliendipoolne keskkond
HTML	<i>Hyper Text Markup Language</i> - veebilehe märgendamise keel
kWp	Tipuvõimsus kilovattides
Low-code	Vähese kasutajapoolse programmikoodiga lahendus
MoSCoW meetod	Prioritiseerimistehnika, mille abil saavad huvigrupid hinnata nõuete olulisust. 4 prioriteeditaset on <i>Must have</i> , <i>Should have</i> , <i>Could have</i> ja <i>Would like</i> .
PDF	<i>Portable Document Format</i> - portitav failivorming
PLC	<i>Programmable Logic Controller</i> - programmeeritav loogikakontroller
REST	<i>Representational State Transfer</i> - veebiteenusega suhtlemise liidese arhitektuur, kus kasutatakse selleks eelnevalt kindlaks määratud ilma olekuta päringuid
SCADA	<i>Supervisory Control And Data Acquisition</i> - juhtimis- ja visualiseerimissüsteem, mis võimaldab koguda, jälgida ja analüüsida seadmete poolt edastatavat infot
SFTP	<i>Secure File Transfer Protocol</i> – turvaline failiedastus protokoll
SMTP	<i>Simple Mail Transfer Protocol</i> - lihtne meiliedastusprotokoll
SQL	<i>Structured Query Language</i> - struktuurpäringukeel andmebaasiga suhtlemiseks
SVG	<i>Scalable Vector Graphics</i> - skaleeruv vektorgraafika
<i>tag</i>	Andmesilt, mis hoiab andmeid ning uuendab neid andmeid vastavalt sisemisele konfiguratsioonile kindla intervalli tagant

# Sisukord

1 Sissejuhatus .....	9
2 Probleemi kirjeldus.....	12
2.1 Varasem lahendus Sunly-s.....	12
2.2 Eksisteerivate päikeseparkide monitoorimislahenduste analüüs .....	14
3 Loodava lahenduse skoop.....	16
3.1 Monitoorimislahenduse nõuete määramine .....	16
3.2 Raporteerimislahenduse nõuete määramine .....	18
3.3 Ajakava paikapane.....	19
4 Tehnoloogia valik.....	20
4.1 Tehnoloogia valiku põhjendus.....	20
4.2 SCADA platvormide võrdlus .....	21
5 Funktsionaalsuse loomine.....	25
5.1 Päikeseparkide monitoorimiseks vajaliku funktsionaalsuse arendamine .....	25
5.2 Raporteerimislahenduse loomine .....	28
5.3 Testimine ning tagasiside .....	31
5.4 Tagasiside põhjal tehtud täiendused .....	32
6 Hinnang rakendatud uuendustele .....	35
7 Ettepanekud edasiseks arenduseks .....	37
8 Kokkuvõte .....	39
Kasutatud kirjanduse loetelu .....	40
Lisa 1. Lihtlitsents .....	42
Lisa 2. Programmikoodi kirjutamine Ignition Designer arenduskeskkonnas.....	43
Lisa 3. Visand monitoorimiskrakenduse vaatest Figma keskkonnas .....	44
Lisa 4. Varahaldurile avanev vaade kõikide päikeseparkidega.....	45
Lisa 5. Kliendi vaade kliendi kasutajaõigustega .....	46
Lisa 6. Varahalduri kasutajaõigustega kliendivaade .....	47
Lisa 7. PowerChart tööriist.....	48
Lisa 8. Hüpinkaken tootmisandmete allalaadimiseks.....	49
Lisa 9. Inverterite staatuse hüpinkaken .....	50
Lisa 10. Alarmist teavitamine kasutades Alarm Notification Pipeline moodulit.....	51
Lisa 11. Kommunikatsiooniprobleemidest teadaandva alarmi seadistamine läbi tag-i .	52

Lisa 12. Kuuraporti raportipõhja esimene lehekülg .....	53
---	----

## Jooniste loetelu

Joonis 1. Kahe erineva andmemudeli loogikastruktuuri lihtsustatud visuaalne representatsioon .....	13
Joonis 2. Kliendi tootmisandmete SFTP serverisse saatmine CSV kujul .....	33



# 1 Sissejuhatus

Peamiselt fossiilsete kütuste põletamisel põhineva traditsioonilise elektritootmisega on seotud suured probleemid nii lühikeses perspektiivis, tänu energiapuudusele ning poliitilistele faktoritele, kui ka pikas perspektiivis, tänu ressursi piiratusele. Kliimamuutuse mõjud on üha ilmsemad ning annavad endast märku aina tihedamini, sundides üleilmset üldsust ning valitsusi vastu võtma erinevaid meetmeid inimtekkeliste kliimamõjude vähendamiseks. Seepärast on taastuvenergiast põhinevatele elektritootmisviisidele üleminek looduskeskkonna aspekte ning sotsiaal-majanduslikku kasu arvestades paratamatu. Päikesepaneelidega või tuulest toodetud elekter on ka soodsaim elektritootmise viis [1]. Efektiivne taastuvenergia allikate kasutamine on aga keeruline, sest seda mõjutavad ilmastikuoludest sõltuvad faktorid. See tähendab, et taastuvenergiaallikad vajavad efektiivseks kasutamiseks lisaks veel teisi tehnoloogilisi lahendusi, mis võimaldavad kontrollida ning jälgida tootlikkust.

Teekonnal rohelisema energiatootmise poole on oluline roll mängida päikeseparkidel, mille kättesaadavus ning paigaldamise lihtsus teevad sellest atraktiivse variandi lisaks suurtele energiatootjatele ka tavainimeste ning entusiastide jaoks. Päikeseelektrijaama rajamine on üks riskivabamaid investeeringuid, olles samal ajal ka suur panus puhtasse elukeskkonda [2]. Lisandväärtusena on võimalik päikesepaneele paigaldada esteetiliselt meeldivate lahenduste abil kohtadesse, kus nad annavad juba kasutatud pinnale lisandväärtust – näiteks katusepleki asemel on võimalik teha katus päikesepaneelidest, mis imiteerivad katusekive.

Päikeseparkide optimaalne paigaldamine nõuab erialaseid teadmisi ning oskusi, mistõttu on just eraklientidel ning väiksematel energiatootjatel mõistlik probleemide vältimiseks valida terviklahendus mõnelt päikesepargi terviklahenduse pakkujalt. Terviklahendus tähendab, et lisaks päikesepargi soovipärasele paigaldamisele aitab teenusepakkuja pargi olukorda monitoorida ning vajadusel viib läbi ettenähtud hooldust ning teostab parandustöid. Päikesepargid nagu enamus hajatootmisel kasutatavad tootmiseseadmed on projekteeritud autonoomsena, inimeste pidev juures viibimine ei ole vajalik ning probleemide märkamiseks ja teatud olukordades ka lahendamiseks piisab haldustarkvara kasutamisest. Terviklahenduse pakkuja on seega hajatootja, kellel on rohkelt kliente ning hallatavaid parke. See tähendab, et efektiivse päikeseparkide töö tagamiseks peab

terviklahenduse pakkuja looma endale sellised tööriistad, mis võimaldaksid just eelmainitud – tegeleda kohapeal ainult nende probleemidega, mis seda realselt nõuavad.

Optimaalse töö tagamine nõuab päikesepargi haldajalt keeruliste seadmete töö monitoorimist ning analüüsimist. Nende töö tulemusena on võimalik vähendada süsteemi rikkeperioodi ning maksimeerida saadavat kasu energiatootmisel. Õigete otsuste tegemine on seega oluline nii tehniliselt, pikendades näiteks seadmete eluiga, kui ka finantsilisel kaalutlustel, tootmisperioodil mitte-operatiivses olekus olek tähendaks tootmata jäänud energiat. Õigeid otsuseid saab aga teha kui nende otsuste aluseks olevad andmed on korrektsed, õigel ajal kättesaadavad ning lihtsasti hoomataval kujul. Automaatjuhtimisseadmetelt ja mõõteseadmetelt - anduritelt ja arvestitelt - info pärimine ning töötlemine toimub energeetikas suurel skaalal SCADA (juhtimis- ja visualiseerimissüsteem, mis võimaldab koguda, jälgida ja analüüsida seadmete poolt edastatavat infot) süsteemide abil. SCADA süsteemid lihtsustavad eelmainitud seadmete ühendamist ning aitavad visualiseerida andmeid. Antud lõputöö kontekstis kasutas lõputöö autor traditsiooniliste veebiarendustehnikate asemel ära modernsete SCADA lahenduste funktsionaalsust luua kasutajasõbralikke kasutajaliideseid.

Lisaks monitoorimisele on SCADA süsteemid kasutusel ka kaugjuhtimiseks [3]. Päikeseparkide terviklahenduse puhul on kaugjuhtimisel eriti oluline roll, sest see võimaldab saata käsked seadmetele ilma, et mõni operaator peaks päikeseparki kohale sõitma ning tegema mõne elementaarse lülituse. Hajatootmise tingimustes on SCADA süsteemid seetõttu väga olulisel kohal.

Lisaks mugavale varahaldusrakendusele on terviklahenduse populariseerimise puhul üks oluline komponent klientidele mugava kasutajakogemuse pakkumine. Kuna päikesepargi paigaldamisega kaasneb märkimisväärne kapitalikulu, on mõisteta, et kuigi suurem töö jääb terviklahenduse pakkuja teha, siis oma investeeringul soovitakse mingil määral siiski jooksvalt silma peal hoida. Kliendid soovivad näha oma päikeseparkide tootmisandmeid väga erineval viisil. On kliente, kes soovivad näha üldiseid andmeid ühe korra kuus, kuid leidub ka kliente, kes soovivad paremate juhtimisotsuste langetamiseks tootmis- ja monitoorimisandmetesse põhjalikumalt süveneda.

Probleemi lahendamisel analüüsi lõputöö raames erinevaid päikeseparkide monitoorimiseks mõeldud veebi- ja mobiilirakendusi ning lisaks analüüsi Eestis konkurentide poolt kasutavaid SCADA lahendusi.

Lõputööna loodav varahaldus- ning raporteerimislahendus sai loodud lõputöö autori poolt olles praktiliselt taastuenergia ettevõttes Sunly. Loodud lahendus on tehtud täpsemalt Sunly kontserni kuuluvale ettevõttele Sunly City-le, mis pakub päikeseparkide terviklahendust era- ja äriklentidele.

## 2 Probleemi kirjeldus

Probleemist ülevaate saamiseks oli vajalik varasemalt Sunly Citys eksisteerinud lahenduse nõrkuste ning tugevuste väljatoomine ning analüüs. Lisaks eksisteeriva lahenduse analüüsile oli vajalik teiste samas sektor eksisteerivate lahenduste analüüs, et paremini aru saada sellest, millised ühised omadused populaarsemaid lahendusi ühendavad. Võrdlusmomendi loomine oli kriitilise tähtsusega projekti skoobist arusaamiseks.

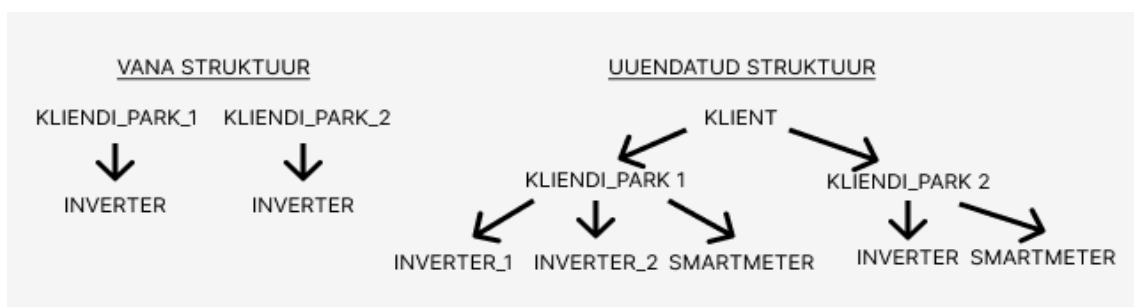
### 2.1 Varasem lahendus Sunly-s

SCADA süsteemide vajalikkus Sunly kontekstis on ilmselge, sest kuigi eksisteerivad tootmiseseadmete tootjapoolsed monitoorimisrakendused ei võimalda need koondada teiste tootjate seadmeid ega luua suuri projekte, kus on vajadus ka spetsiaalsete lahenduste lisamiseks. Sunly kasutab enda päikeseparkide kaugjälgimiseks ning kontrollimiseks Ignition Automation tarkvara.

Enne lõputööd eksisteerinud lahendus era- ning äriklientide päikeseparkide töö monitoorimiseks oli loodud kiirkorras Sunly City eksisteerimise algusfaasis. Olemasoleva lahenduse puhul oli tegemist Ignition SCADA platvormil Perspective mooduliga loodud kiiret ülevaadet pakkuva SCADA vaatega. Ignition Perspective on Ignition tarkvara lisamoodul, mis võimaldab luua täisfunktsionaalseid tööstuslikke rakendusi, mille *frontend* (kasutajaliides, kliendipoolne keskkond) on HTML5-s (veebilehe märgendamise keel), mille elementide kujundust on võimalik muuta läbi CSS-i (veebilehtede küljendamisel kasutatav märgistuskeel) ja *backend* (teenusepoolne keskkond) loogikat saab kirjutada Jythonis [4]. Olemasolev lahendus oli aga ajast maha jäänud, sest sinna vaatesse oli ühendatud vaid paar päikeseparki ning uute lisamine süsteemi oli keerukas ning aeganõudev. Varasema lahenduse näol puudus tegelikult seega monitoorimisrakenduse funktsionaalsus, kust mõni operaator oleks saanud kõikide parkide monitoorimiseks vajamineva informatsiooni.

SCADA süsteemi uute päikeseparkide lisamine toimus skeemi järgi, et loodi päikesepargi objekt, millega ühendati ära inverterite olemid. Inverterite olemit ei olnud varasemalt aga loodud loogikat, mis viiks infot ülemise taseme ehk pargi objekti. Kokkuvõtvalt oli struktuuris probleeme alumise taseme objekti info edasi andmisega ülemisele. Sunly City

kasvades võttis arenduse üle uus SCADA arendaja, kes kirjutas info süstematiseerimiseks uue loogika. Uue loogika kohaselt eksisteeris kolm omavahel seotud elementi: klient, park ja seade. Uuendatud loogika oli vajalik eelkõige sellepärast, et Sunly City-s hakkas tekkima kliente kellel oli mitu päikeseparki ning selline loogikamuutus aitas kaasa süsteemis struktuuri loomisele. Lisaks tekkis eelmainitud uutesse parkidesse peale inverterite ka teisi seadmeid ehk vana loogikaga ei saanud enam edasi jätkata tegelikult juba ainuüksi selle muutuse tõttu. Uue andmestruktuuriga tekkis informatsiooni liikumine alumise taseme elemendilt, seadmelt, ülemistele struktuurielementidele, päikesepargile, mis lihtsustas märkimisväärselt infopäringuid mitme seadmega päikeseparkides. Näiteks saab siinkohal välja tuua ühe olulisema näitaja ehk kuidas esmalt saadetakse kõikide inverterite poolt toodetav energia pargi infosse ning seejärel saadetakse pargi tootlus edasi kliendi tasemele. Selline info liikumine lihtsustas märkimisväärselt päringuid, mida pidi tegema tootmisandmete saamiseks ning mis veelgi olulisem, eelmainitud muutus kiirendas andmepäringuid, mis on lõppkasutaja kasutajakogemuse vaatenurgast suur pluss. Joonisel 1 on lihtsustatud kujul illustreerimiseks välja toodud kahe erineva struktuuri erinevused.



Joonis 1. Kahe erineva andmemudeli loogikastruktuuri lihtsustatud visuaalne representatsioon

Kuigi uus struktuur lihtsustas andmepäringuid, siis ei muutunud olukord monitoorimiskanduse vaatenurgast. Uut struktuuri ei saanud ära kasutada, sest eelnevalt loodud vaated olid loodud väga järgalt esialgsele loogikale. Probleemi allikaks oli fakt, et Ignition SCADA platvormil on struktuuri jälgimine Ignitioni enda sisseehitatud tööriistadega infopäringute tegemiseks väga tähtis ning kui puudub vahekiht erinevate struktuuride ühtlustamiseks, siis lõpetavad SCADA vaated töötamise. Lisaks eelmainitule puudus loogika, mis võimaldaks lihtsalt kokku liita kõikide Sunly City päikeseparkide monitoorimiseks vajaminevaid andmeid.

Eelmainitud rakenduses puudus raporteerimislahendus täielikult, kuigi see oli midagi, mida oli Sunly City oma klientidele lubanud pakkuda. Kuna Sunly peamine

sissetulekuallikas ei ole Sunly City poolt eraklientidele pakutav terviklahendus, siis ei olnud varahaldus ja raporteerimislahenduse loomine pikalt Sunly SCADA-t arendava meeskonna jaoks prioriteediks. Kuid olukorras, kus klientide arv oli võrreldes algse lahendusega kümnekordistunud, tekkis vajadus uue lahenduse loomiseks.

## **2.2 Eksisteerivate päikeseparkide monitoorimislahenduste analüüs**

Eksisteerivate lahenduste vastu võrdlemine sai toimuda vaid klientidele pakutavate võimaluste põhjal. Suuremate energiafirmade, mis tegelevad samuti hajatootmisega, kasutatavatele süsteemidele ei olnud võimalik lõputöö autoril ligipääsu saada. Kuidas nad kaugjälgimist teostavad on selline siseinfo, millele juurdepääsu saamine oleks aidanud analüüsi faasis palju. Õnneks oli lõputöö autoril ligipääs kahele populaarsele päikeseparkide monitoorimiskandele, mis on suunatud tavakasutajatele. Kahe sellise monitoorimiskandele olemasolu võimaldas teha analüüsi peamiste funktsionaalsuste tugevuste ning nõrkuste kohta.

Esimene firma, mille lahendus sai analüüsi aluseks võetud, oli Huawei. Huawei näol on tegemist suure tegijaga globaalsel tasandil, mistõttu oli nende pakutav lahendus väga hea inspiratsiooniallikas loodava rakenduse potentsiaalsete elementide jaoks. Huawei on välja töötanud keskkonna FusionSolar, millele pääseb ligi nii nutitelefonist kui ka veebibrauseri kaudu. FusionSolaris näeb nii reaalaaja- kui ka ajaloolist päikesepargi tootlust, seadmete olukorda, hetkeilma päikesepargi asukohas kui ka seda kui palju energiat liigub võrku, salvestusseadmetesse ning kliendile endale. Kõike seda antakse edasi ühe veebivaate vahendusel ehk ei ole vaja ülevaate saamiseks liialt navigeerida. Lõputöö autori arvates on rakendusse lisatud ka informatsiooni, mis on üleliigne. Nimelt on seal väljatoodud informatsiooni, mida on küll tore lugeda aga väärtus monitoorimise ning tootlikusele hinnangu andmise vaatenurgast on olematu. Sellise ebavajaliku informatsiooni alla kategoriseerub lõputöö autori arvates näiteks see, et oli väljatoodud mitu puud selle päikesepargiga päästetud on või mitu kilogrammi süsihappegaasi jäi atmosfääri paiskamata tänu toodetud puhtale energiale. Kuigi, nagu sai eelnevalt mainitud, need numbrid võivad mõne kliendi jaoks huvitavad olla, oleks lõputöö autori arvates võinud seda ruumi brauserivaates mingi muu, kasulikuma infoga täita.

Kokkuvõtvalt oli Huawei lahendus lõputöö autori arvates hea, kuid selle lahenduse puhul oli kohti, mis jätvad soovida hajatootmise vaatenurgast. Üks probleem, mis jäi

FusionSolaris silma, oli kokkuvõtva vaate ning kokkuvõtivate andmete puudumine. Lisaks puudus alarmide ajaloolise ülevaate olemasolu ehk kuigi kõikidest päikeseparkidest oli väga hea ülevaade loodud nii ajalooliste kui reaalajas andmetega, puudus vaade, mis selle kõik omakorda kokku võtaks. Selline vaade on aga varahaldaja vaatenurgast väga väärtuslik.

Teine võrdluseks võetud lahendus oli mobiilirakendus, mille on loonud oma seadmete jaoks AlphaESS. Kuigi AlphaESS pakub samuti nii veebi- kui ka mobiilirakendust, siis teiseks analüüsi aluseks sai valitud mobiilirakendus. Lõputöö autor tegi sellise otsuse, sest nii tekkis hea võimalus luua võrdlusmoment veebibrauseri, kus on palju ruumi erineva info esitlemiseks, ning mobiilirakenduse, kus on ruumi vähem ning tuleb hoolikalt valida mis infot kliendile näidata, vahel. AlphaESS lahenduse uurimisel sai selgeks, et päikesepargi monitoorimiseks mõeldud rakenduse puhul, mis on mõeldud operaatorite ja varahaldurite asemel tavaklientidele, on tegelikult vaja näidata vaid paari olulist infokildu. Funktsionaalsust oli rakendusse loodud küll palju, kuid pargi seisukorra hindamiseks piisas vaid kiirest pilgust avakuvale.

Uuritud AlphaESS lahenduse puhul oli sarnaselt Huawei lahendusele puudu aga võimalus monitoorida suurt hulka päikeseparke ehk hajatootmise puhul muutuks varahalduri töö ebamugavaks ning keeruliseks.

## 3 Loodava lahenduse skoop

Loodava lahenduse skoobi defineerimisel oli oluline jagada projekt kaheks ja keskenduda nende nõuete määramisele kui kahele täiesti erinevale projektile. Kuna monitoorimisrakenduse puhul hakkavad seda kasutama nii kliendid kui ka firmasisesed isikud tuli ka see projekti osa nõuete määramise jaoks kaheks lüüa, sest erinevus vastavalt kasutajatuübile oli midagi, mis pidi kindlasti eksisteerima.

### 3.1 Monitoorimislahenduse nõuete määramine

Võrreldes Huawei FusionSolariga ning AlphaESS monitoorimisrakendusega, mille kallal töötavad suured arendajate tiimid, töötas lõputööna teostava projekti kallal üks arendaja. Sellepärast tuli tehtavale tööle seada realistlikud ootused, samas luues siiski ajapiiranguid arvestav Sunly Cityle lisandväärtust pakkuv lahendus, mis lahendab väga aktuaalset probleemi.

Skoobi paikapanek algas erinevate osapoolte ja nende huvide ära kaardistamisest. Loodava rakenduse puhul olid osapoolteks Sunly City varahaldurid, projektijuhid, erakliendid, ning ärikliendid. Lõputöö autor kirjeldas kohtumistel funktsionaalsust, mida võiks loodavasse rakendusse luua põhinedes varasemale Huawei ja AlphaESS lahenduste analüüsi käigus saadud informatsioonile. Arutelu tulemusena ning kokkupanud informatsiooni põhjal algas protsess kasutajalugude loomiseks. Kasutajalugude kogumiku komplekteerimisele järgnes kasutajalugude analüüs, mille käigus sai rakendatud MoSCoW meetodit (prioritiseerimistehnika, mille abil saavad huvigrupid hinnata nõuete olulisust. 4 prioriteeditaset on *Must have*, *Should have*, *Could have* ja *Would like*) analüüsiks, et defineerida erinevate kasutajalugude poolt loodava funktsionaalsuse prioriteet arenduseks.

Peale esialgset skoobi paikapanekut toimus veel IT tiimi sisene arutelu, kuidas oleks kõige parem soovitud rakendust luua ning milliseid õppetunde oli saadud varasemate arenduste käigus. Kõige olulisemana selgus, et parkide süsteemi lisandumisel peaks nende vaadetesse ning kokkuvõtvatesse andmetesse lisandumine toimuma võimalikult dünaamiliselt. Dünaamilisus ei olnud varasemalt kriitilise tähtsusega, sest enne Sunly City eksisteerimist tegeleti Sunly-s vaid suurte päikeseparkide projektidega, kus on vähe seadmeid aga suured võimsused. Sunly City-s on aga vastupidi, seadmeid ning



päikeseparke on palju aga võimsused on väiksed. See tähendas, et kui varasemalt oli võrrandeid käsitsi tehtud, siis uue versiooni puhul tuli liikuda võrrandite kirjutamisele läbi programmikoodi. Programmikoodi puudumine varasematest lahendustest tulenes suuresti sellest, et esmase rakenduse väljatöötanud arendaja ei omanud väga palju kogemust programmeerimises ning kuna seadmete ja päikeseparkide arv oli väike, siis oli ajaliselt optimaalne arendada nii, et suur osa tööst, näiteks *tag*-i (andmesilt, mis hoiab andmeid ning uuendab neid andmeid vastavalt sisemisele konfiguratsioonile kindla intervalli tagant) SCADA vaate elemendiga sidumine, toimus käsitsi. Lahenduse prototüüpimiseks on selline lähenemine kindlasti turvalisem ning kiirem, kuid skaleerimiseks oli vajalik teha muudatus programmikoodi ning dünaamilisuse poole. Ignition SCADA platvorm pakub programmikoodi kirjutamiseks sisseehitatud keskkonda (vt Lisa 2) ning kuigi platvormil on kasutusel Jython programmeerimiskeel, on pärast dokumentatsiooni läbitöötamist ning sisseehitatud funktsioonide äraõppimist päris mugav uusi lahendusi läbi programmikoodi luua.

Kasutajalugudest parema ülevaate saamiseks grupeeris lõputöö autor need pärast prioriteerimist kolme erinevasse kategooriasse: reaajas monitoorimine kliendile ja varahaldajale, ajalooliste andmete ülevaade kliendile ja varahaldajale ning raporteerimislahendus kliendile. Võttes aluseks kahes analüüsi aluseks olnud rakenduses nähtu ning kombineerides see teiste SCADA arendajate kogemustega jõuti skoobi paikapanekul järeldusele, et ideaalis võiks rakendus olla sarnaselt eelmisele lahendusele ühe-lehe veebirakendus. Ühe-lehe veebirakendus on varahalduse seisukorrast väga sobiv lahendus, sest navigeerimine erinevate vaadete vahel ei ole vajalik seni kuni kõik otsuste tegemiseks ning päikeseparkide töö hindamiseks vajaminevad andmed on võimalik ühe vaatega edasi anda. Analüüsi ning arutelu tulemusena jõuti otsusele, et vaade võiks olla tehtud nii, et selle saaks jätta kuhugi televiisorile või monitorile jooksma ning juba ainuüksi nii peaks olema võimalik saada ülevaade toimuvast. Päris ilma nupuvajutusteta ei olnud aga võimalik rakendust luua, sest lisaks kõikidest parkidest ülevaadet andvale vaatele oli näiteks vaja ka võimalust filtreerida klientide ja parkide vahel ning teha ajaloolisi päringuid kasutaja poolt defineeritud ajaperioodi alusel.

Suurim muutus võrreldes varasema lahendusega pidi tulema päikeseparkide eest vastutavatele isikutele alarmide edastamise vaatenurgast. Monitoorimisrakendus on hea olukordades, kus see on saadaval, aga kindlasti tekib olukordi, kus vastutav inimene on monitoriekraanist eemal või tegeleb muude ülesannetega. Sellises olukorras oleks teda

siiski vaja teavitada tekkinud probleemidest. Põhinedes teiste lahenduste analüüsile ning funktsionaalsuste kaardistamise faasis kuuldule, sai paika pandud, et välja tuleks hakata saatma teavitusi nii olukorras, kus inverter või mõni muu seade saadab veakoodi, kui ka olukorras, kus side pargiga on pikema perioodi katkenud. Neist kahest hilisem on hajatootmise vaatenurgast väga oluline, kuid siiski selline funktsionaalsus, mida teistes analüüsi käigus uuritud lahendustes polnud. Teavitus kommunikatsiooniprobleemide kohta on vajalik, sest võib tekkida olukord, kus päikesepargis on tegelikult kriitiline alarm, kuid side katkemise tõttu ei saa sellest keegi teada.

Rakenduse funktsionaalsus pidi hakkama erinema ka vastavalt kasutaja rollist. Lisaks sellele, et varahaldajatel on ligipääs kõikide süsteemi päikeseparkidele vajadid varahaldajad ka rakenduses ligipääsu Ignitioni tööriistale PowerChart, mis võimaldab uurida veelgi põhjalikumalt seadmete poolt tulevat informatsiooni. PowerChart tööriist on probleemide põhjuste allika väljauurimiseks väga võimas tööriist, kuid selle kasutamise keerukuse tõttu ei olnud seda mõistlik era- ega ka äriklientidele pakutava funktsionaalsuse hulka lisada.

### **3.2 Raporteerimislahenduse nõuete määramine**

Raporteerimislahenduse nõuete määramisel kasutas lõputöö autor ära konkureerivate lahenduste analüüsi osas välja uuritud ehk suuresti kattub väljasaadetavas raportis esitatud informatsioon nende punktidega, mida sai varasemalt ka monitoorimISRakenduse puhul ära kaardistatud. Peale arutelu ning koosolekuid erinevate osapooltega jõuti järeldusele, et on nüansse, mida raporti puhul tuleb juurde lisada. Raporti puhul sai planeerimisfaasis välja toodud, et kui monitoorimisvaate graafiline pool võib olla lihtne, sest eesmärgiks ei ole, et rakendust oleks ilus vaadata vaid, et andmed oleks kiirelt ning mugavalt kättesaadavad, siis raporti puhul tuleb tõsisemalt tähelepanu pöörata disainile ning sõnastusele. Lisaks tuli nõuete määramise faasis välja, et oluline oleks anda klientidele mingisugune mõõdupuu erinevate kuude võrdlemiseks. Selline nõue tulenes faktist, et olukorras, kui on väga päikesepaiste vaene kuu, siis ei ole ka tootlikus eriti suur ning klient vajab sellest arusaamiseks võrdlusmomenti. Sellepärast sai raporti nõuete hulka lisatud see, et võrdlusmomenti loomiseks tuleb raportisse viimase kahekümne aasta info põhjal raporteeritava perioodi keskmine päikese- ning vihmainfo. Näiteks olukorras, kus raporteeritava perioodi jooksul oli 80 päikesetundi, aga 20 aasta keskmine sellel samal

perioodil oleks 100 päikesetundi, siis saaks näidata, et võrreldes pikaajalise keskmisega oli päikesetunde 80%. Väljasaadetavas raportis sooviti lisaks näha ka lõputöö autori poolt varasema Huawei ning AlphaESS lahenduste analüüsi ning võrdluse käigus väljatoodud elemente. Üks selliseid elemente oli informatsioon selle kohta kui palju oli tänu antud kliendi päikeseparkidele vähem süsinikdioksiidi atmosfääri paisatud. Lisaks tuli raportisse sisse tuua kaotatud toodangu hulk ehk energiahulk, mis jäi tootmata mõne päikesepargis oleva seadme rikke tõttu.

Suure hulga klientide tõttu tuleb välja saata ka suur hulk raporteid ehk ka see oli midagi mida sooviti automatiseerida.

### **3.3 Ajakava paikapane**

Lõputöö ajakava paikapane kul lähtus lõputöö autor agiilse arenduse meetoditest. See tähendas kahe nädalasi sprinte, kus igaks tsükliks sai välja valitud mõned funktsionaalsused. Funktsionaalsuste arendamise järjestamisel jälgis lõputöö autor funktsionaalsustele MoSCoW analüüsi käigus antud hinnangut. Siiski tekkis olukordi, kus oli mõistlik vahele võtta ka madalama prioriteediga ülesandeid. MoSCoW analüüsist oli abi, sest see aitas suures määras arendust planeerida ning andis iga mööduva nädalaga enesekindlust, et projekt püsib ajakavas. Projekti ajakavas püsimist mõjutab fakt, et lõputöö autor leidis arenduse käigus mitmel korral uusi Ignitionisse sisseehitatud funktsionaalsusi, mille järel sai juba varasemalt tehtud funktsionaalsust optimeeritud.

## 4 Tehnoloogia valik

Tehnoloogia valiku protsessil pidi lõputöö autor lähtuma mitmest faktorist. Sunly-poolne sisend tehnoloogia valikul oli, et automaatika töökindluse tagamiseks oleks SCADA platvormi kasutamine mõistlik otsus. Lõputöö autor on aga õppinud selliseid rakendusi looma kasutades traditsioonilisi veebiarendustehnikaid. Peale loodava lahenduse skoopi ning ajakava defineerimist oli hea võrrelda mõlema poole, traditsiooniliste veebiarendustehnikate ning SCADA platvormide, tugevusi ning nõrkusi, mille põhjal teha otsus kasutatavate tehnoloogiate ning meetodite osas.

### 4.1 Tehnoloogia valiku põhjendus

Traditsioonilised SCADA-süsteemid on suured ja aeglased ning lisafunktsionaalsuse saavutamiseks peab tihtipeale kombineerima erinevaid tooteid ning teenuseid. Selline kokkusobitamine tähendab, et need süsteemid võivad osutada keerulisteks lahendusteks, mis ei pruugi alati sobida konkreetsete kasutusjuhtude ega kliendi äri vajadustega, tekitades seeläbi erinevaid probleeme lahenduste arendamisel ning pikaajalisel kasutamisel. Kohandatud muudatused võivad sageli kaasa tuua tarkvarakonflikte ja tundlikkust põhiliste hooldustööde suhtes, näiteks uuenduste tegemisel [6]. SCADA platvormid on aga teinud viimase kümne aastaga suuri edusamme, olles nüüdseks jõudnud punkti, kus neid saab kasutada ka rakenduste loomiseks, mida tavaliselt oleks arendatud tavapärase veebiarendustehnikatega [7]. Antud lõputöö kontekstis kasutaski lõputöö autor traditsiooniliste veebiarendustehnikate asemel modernset SCADA lahendust. Selline otsus sündis, sest kuigi traditsiooniliste veebiarendustehnikate kasutamine võib pakkuda suuremat kontrolli ja paindlikkust, pakuvad kaasaegsed SCADA lahendused, nagu Inductive Automation Ignition SCADA, mitmeid eeliseid, mis võivad kiirendada arendusprotsessi ja vähendada üldisi arenduskulusid. Lisaks on need loodud töötama tööstusautomaatika kontekstis, pakkudes tööriistu ja funktsioone, mis on spetsiaalselt loodud selle sektori vajaduste rahuldamiseks.

SCADA platvormidel on traditsiooniliste veebiarendustehnikate ees lõputöö autori analüüsi põhjal kolm peamist eelist. Esiteks, SCADA platvormidel on sisseehitatud PLC (programmeeritav loogikakontroller) loogika ehk nad on loodud töötama koos programmeeritavate loogikakontrolleritega, mis on tööstusautomaatika

põhikomponendid. Tänu sellele suudavad SCADA süsteemid suhelda otse PLC-tega ja kasutada nende loogikat. See tähendab, et arendaja ei pea muretsema oma *backend* süsteemi loomise pärast loogikakontrolleritega suhtlemiseks [8]. Teiseks, oluline faktor on siltide ehk *tag*-ide funktsionaalsus. SCADA süsteemides kasutatakse *tag*-e reaajas väärtuste esindamiseks, näiteks sensori oleku või inverteri aktiivvõimsuse jälgimiseks. Sildid ehk *tag*-id on muutujad, mis suudavad salvestada andmeid ja on kasutajate poolt võimalik konfigurida lisaks sensorite väärtuste näitamisele näiteks andmebaasi päringu tulemuste kui ka kasutaja poolt defineeritud väärtuse hoidmise jaoks [9]. Sellise kasuliku funktsionaalsuse olemasolu tähendab, et arendaja peab andmehaldusloogika loomise jaoks vähem vaeva nägema. Kolmandaks, SCADA eelis traditsiooniliste veebiarendustehnikate ees on arenduse kiirus. Modernsed SCADA lahendused pakuvad juba olemasolevaid tööriistu, komponente ja funktsioone, mis aitavad kiirendada ja lihtsustada kasutajaliidese loomist. SCADA lahenduste, nagu näiteks Sunly poolt kasutatav Inductive Automation Ignition SCADA, puhul on sellepärast võimalik läbi saada märgatavalt vähesema programmikoodiga. *Low-code* (Vähesese kasutajapoolse programmikoodiga lahendus) ja koodivaba kasutajaliidese arendamine on üha enam tunnustatud kui veebiarenduse tulevik. Haseeb Tariqi Forbesi artikkel “Low-Code Versus No-Code And The Future Of Application Development” toob välja, et *low-code* ja koodivabad platvormid võimaldavad arendajatel luua rakendusi kiiremini ja tõhusamalt [10].

## 4.2 SCADA platvormide võrdlus

Kiire kasvuga ettevõttes on oluline omada vastupidavaid, paindlikke ja tõhusaid süsteeme. Selliste süsteemide tunnusjooned on skaleeritavus, mitmekülgsus, kasutajamugavus ja modulaarsus, mis on kõik kriitilise tähtsusega pikaajaliseks eduks ja innovatsiooniks [11]. Skaleeritavus tagab, et süsteemid töötavad mitte ainult praegu, vaid ka tulevikus, kus sama lahendus peab suutma laienedes üleval hoida palju võimsamat infrastruktuuri. See tähendab, et SCADA süsteemide puhul peab süsteem olema valmis kohanema suurenevate andmemahtudega, uute seadmete lisamisega või isegi kogu struktuuri ümberkorraldamisega. Mitmekülgsus aga võimaldab süsteemil kohaneda uute tehnoloogiatega, tagades seeläbi selle pikaajalise elujõulisuse. See tähendab, et süsteem suudab vastu võtta tulevasi uuendusi ja integratsioone, olles alati ajakohane.

SCADA lahendusi on palju, mistõttu on oluline võrrelda erinevate SCADA platvormide nõrkusi ja tugevusi ning seeläbi põhjendada, miks lõputöö autor otsustas just Ignition Automation SCADA kasuks. Käesoleva analüüsi aluseks võetakse kolm SCADA lahendust: Ignition Automation SCADA, Aveva Wonderware ja GE PowerOn Reliance. Aveva Solutions, endise nimega Aveva Wonderware, lahendust kasutab Enefit Green ning GE PowerOn Reliance oli pikaajaline tööstuspartner Eleringi jaoks. Võrdluse raames tuleb vaadata erinevatele lahendustele peale tugevalt ka veebiarenduse vaatenurgast, sest eesmärgiks on kasutada ära modernsete SCADA lahenduste pakutavat justnimelt traditsiooniliste veebiarendustehnikate asemel.

Ignition SCADA, mis on välja töötatud kasutades kaasaegseid tarkvaratehnoloogiaid nagu pilvepõhised lahendused, avatud standardite toetamine, REST (veebiteenusega suhtlemise liidese arhitektuur, kus kasutatakse selleks eelnevalt kindlaks määratud ilma olekuta päringuid) API-de (rakendusliides) integreerimine ja veebipõhised kasutajaliidesed, on kujunenud üheks juhtivaks lahenduseks tööstusautomaatikas. Üks Ignition SCADA märkimisväärseid omadusi on selle modulaarne ja skaleeritav lähenemine, mis ei ole mitte ainult paindlik vaid ka majanduslikult efektiivne. See võimaldab ettevõtetel mugavalt lisada uusi servereid, optimeerida süsteemi jõudlust ja suurendada süsteemi kättesaadavust. Seda kõike on võimalik teha ilma, et oleks vaja teha suuri muudatusi olemasolevasse infrastruktuuri. Lisaks on Ignition SCADA platvormil piiramatult arv *tag*-e. See on midagi mis võrreldes järgnevas paragraafides väljatoodud lahendustega on selgelt üks Ignition SCADA lahendust heast küljest eristav omadus. Selline dünaamiline skaleerimisvõime toetab kasvavate organisatsioonide vajadusi, võimaldades neil kiiresti ja tõhusalt kohaneda muutuvate tootmisvajadustega. Ignitioni paindlik litsentsimismudel võimaldab lisada uusi seadmeid ja servereid ilma ettearvamatute kulutuste või keerukusteta. Platvormi õppimise vaatenurgast on Inductive Automationi pakutav lahendus teistest analüüsi käigus analüüsitud lahendustest parem. Inductive Automation foorum on aktiivne ning vastuseid saab kiirelt nii platvormispetsiifilistele kui ka tavalistele probleemidele, näiteks programmikoodi vigadele. Lisaks foorumile eksisteerib ka õppekeskkond Inductive University, mille tasuta kursused teevad sellest väga hea lahenduse, mida kasutada uute tehnoloogiatega süsteemi integreerimisel ning uute arendajate koolitamisel. Ignition kasutab kronoloogiliste andmete kogumiseks SQL-i (struktuurpäringukeel andmebaasiga suhtlemiseks). Selleks kasutatakse PostgreSQL andmebaasi koos abstraktsioonikihiga

nimega Timescale. Timescale on mõeldud suurte kronoloogiliste andmehulkade haldamiseks, jagades andmebaasi automaatselt partitsioonideks ja säilitades jõudluse kuni 1 miljardi reani. Sellise abstraktsioonikihi olemasolu on oluline, et lihtsustada päringute tegemist välisallikatest kasutades SQL keelt, mis on modernse veebiarenduse jaoks väga vajalik lisandväärtus, mis on midagi mida antud lõputöö kontekstis läheb kindlasti vaja.

Aveva Solutions kasutab ArcestrA tehnoloogilist raamistikku, mis on loodud tööstusautomaatika süsteemide integreeritud ja skaleeruva lahendusena. ArcestrA ühendab tööstusautomaatika ja IT, pakkudes platvormi, mis hõlmab kõike alates anduritest kuni ettevõtte ressursiplaneerimise süsteemideni. ArcestrA põhineb objektipõhisel struktuuril, mis teeb automaatikarakenduste loomise efektiivsemaks. Selline struktuur soosib komponentide taaskasutust, aidates seeläbi vähendada arendusega seotud kulusid ja hõlbustada erinevate süsteemide lõimimist. Integreeritud arenduskeskkonna olemasolu võimaldab eri tööstusrakendustel omavahel sujuvalt suhelda, optimeerides seeläbi erinevate süsteemide ja seadmete vahelist koostoimimist [12]. Sarnaselt Ignition SCADA platvormile pakub ka Aveva tugimaterjali, kuid see ei ole nii laiaulatuslik kui Ignition Automation poolt pakutav ning lisaks on paljud Aveva õpperessursid tasulised. Aveva erineb veel Ignition platvormist seeläbi, et see piirab kasutajate ning *tagi*-ide arvu, mis on mõlemad arendatava rakenduse kontekstis olulised väärtused. Lõputöö raames loodava lahenduse kontekstis on Aveva Solutions pakutaval lahendusel üks tugev puudus ning see on fakt, et Aveva platvorm kasutab kronoloogilisi andmebaase ilma Ignition SCADA platvormil nähtud SQL abstraktsioonikihita.

GE PowerOn Reliance pakub rohkem kui ainult SCADA funktsionaalsust. Nende lahendus hõlmab ka elektri tootmise, edastamise ja jaotamise rakendusi elektrivõrkudes [13]. Selline lai funktsionaalsus võib aga muuta seadistamise keerukamaks ning seda eriti arendajatele ja ettevõtetele, kes vajavad ainult SCADA lahendust. Integreerimisprobleemidest ning kasutajasõbraliku rakenduse puudumisest annavad märku ka kriitilised arvustused. Kõnekaim näitaja on see, et vaid 50% kasutajatest soovitaks GE PowerOn Reliance platvormi [14]. Suurimaks probleemiks on aga lõputöö autori arvates see, et õppematerjalid ning muu lahenduste integreerimiseks vajaminev informatsioon ei ole lihtsasti kättesaadav. Lõputöö autor leidis vaid tasulisi koolitusi ning ka tavadokumentatsiooni ei olnud lihtne leida. Platvormi puudujääkidest annab märku

veel see, et Elering alustas 2022. aastal uue platvormi otsingut. Lõputöö valmimise hetkeks ei ole avalikustatud, millisele platvormile Elering liikumas on.

Sooritatud analüüsi põhjal osutus parimaks valikuks antud lõputöö nõudeid ning Sunly ja Sunly City plaane arvesse võttes Ignition Automationi lahendus. Analüüsi tulemus oli selline, sest võtmeküsimustes, milleks on antud juhul kasutatavus veebiarenduseks, skaleeritavus, kasutajasõbralikkus ning hind, oli nende lahendus teistest üle ning kuna antud projekt oli lõputöö autori esimene kokkupuude SCADA süsteemidega, siis oli Inductive University õppeplatvormi olemasolu väga hea lisaväärtus. Lisaks eelmainitule oli Ignitioni õppeplatvorm suureks abiks mõistmaks, kuidas töötavad Modbus ja OPC UA standardid. Ignitioni platvorm kasutab mõlemat protokollit, Modbusi ja OPC UA-d, et tagada erinevate seadmete ühenduvus ja tõhus andmetöötlus.



## 5 Funktsionaalsuse loomine

Funktsionaalsuse loomine jagunes nelja erinevasse faasi. Esimesed kaks faasi olid varasemalt defineeritud skooopi põhjal funktsionaalsuse arendamine. Funktsionaalsuse arendamisele järgnes testimise ning tagasisidefaas, kus erinevad osapooled said kasutada ning anda tagasisidet loodud lahendusele. Viimane samm funktsionaalsuse loomise protsessis oli saadud tagasiside põhjal täienduste tegemine.

### 5.1 Päikeseparkide monitoorimiseks vajaliku funktsionaalsuse arendamine

Firmasiseseks kasutamiseks vajaliku funktsionaalsuse arendamine oli käesoleva lõputöö puhul peamine eesmärk, mistõttu oli just see kogu arenduse alguspunktiks. Kõigepealt oli vaja selgeks saada suurem pilt, milline võiks tulevane varahaldusvaade graafiliselt välja näha, sest juba varasemas arenduse faasis sai ära kaardistatud millised näitajad peavad vaates kajastuma. Lõputöö peamise funktsionaalsuse ehk monitoorimisvaate puhul rakendas lõputöö autor nelja kasutajaliidese loomise printsiipi. Printsiipide aluseks oli Steve Krugi raamat „Don't Make Me Think: A Common Sense Approach to Web Usability.“ [15]. Eelmainitud raamat sai valitud põhjusel, et uurides kasutajaliidese kasutatavuse analüüsi kohta oli antud teos pidevalt üheks allikaks, mis oli aluseks võetud. Kuigi raamatus oli välja toodud rohkem kui neli faktorit, mida jälgida hea kasutajaliidese loomisel, siis valituks osutusid just need, mida lõputöö autor nägi, et on võimalik ära kasutada ka SCADA vaate loomisel. Esimene printsiip tuli raamatu esimesest peatükist ning see oli, et kasutajaliideseid peaksid olema selged ja otsekohesed ehk kasutajad peaksid suutma esitatavat teavet mõistma ilma, et nad peaksid liiga palju mõtlema või läbima ulatuslikku koolitust. Teiseks, teises peatükis sai välja toodud, et võimaluse korral tuleks vähendada eesmärgi saavutamiseks vajalike klõpsamiste või interaktsioonide arvu. Kolmanda printsiibi, mille leiab kuuendast peatükist, puhul oli märgitud, et kriitilist teavet ja olulisi elemente tuleks taustaga võrreldes esile tõsta. See võib tähendada nende keskset paigutamist, eristuvate värvide kasutamist või muid disainivihjeid, et tagada, et need püüavad kiiresti kasutaja tähelepanu. Viimane lõputöö raames kasutatav printsiip rõhutab, et vaateid tuleb kindlasti testida tegelike kasutajate peal. Isegi lihtsad testid, kus jälgitakse mõne inimese liidese kasutamist, võivad paljastada suuri kasutatavusprobleeme, mis ei pruugi disaineritele ilmsed olla.

Peale disainiprintsiipide paikasaamist tulevases SCADA vaatest parema pildi saamiseks sai loodud lihtne visand Figma keskkonnas (vt Lisa 3), et saada kiire esmane tagasiside. Figma on veebipõhine graafilise disaini ja kasutajaliidese loomise tööriist, mis aitab kiiresti liikuda edasi kasutajakogemuse disaini prototüüpimise faasist.

Rakenduse visuaalse poole loomisel kasutas lõputöö autor suures osas Ignitioni enda sisseehitatud komponente, mis lihtsustasid ning kiirendasid arendusprotsessi meeletult. Põhiliselt kasutas lõputöö autor klassikalisi kasutajaliidese komponente nagu rippmenüüd, nupud ning erinevad graafikud ja mõõdikud (vt Lisa 4). Sarnaselt klassikalisele HTML-ile saab ka Ignition Perspective moodulis kasutada CSS-i visuaalse poole parandamiseks, kuid seda sai kasutatud minimaalselt.

Monitoorimisvaate loomisel oli olulisel kohal kasutajaõigustest tulenevalt kasutajale erineva funktsionaalsuse pakkumine. Kuigi suures plaanis on mõlema vaate poolt pakutav informatsioon sama ei ole kliendile mõeldud vaates (vt Lisa 5) kõiki funktsionaalsusi ning filtreerimisvõimalusi, mis on varahaldurile suunatud monitoorimisvaates (vt Lisa 6). Näiteks ei ole kliendil ligipääsu tööriistale PowerChart (vt Lisa 7), mille saab varahaldur avada vajutades aktiivvõimsus kuvavale nupule vaate ülaosas, sest antud sisemise tööriista kasutamine on keeruline ning annaks kliendile ligipääsu ka teistele funktsionaalsustele, mida talle ei soovita pakkuda.

Lõputöö autor oli väga rahul Ignition Perspective moodulis pakutavate veebiarenduseks mõeldud elementidega. Pakutavatest komponentidest kõige problemaatilisem oli kaardi komponent, millel tuuakse välja parkide asukohad. Kaardile markerite loomine ning nende klõpsatavaks tegemine ning pidev uuendamine vastavalt pargi staatusele osutus üheks lõputöö problemaatilisemaks osaks. Kuigi Ignitioni kaardikomponent kasutab Leaflet JavaScript teeki, mis on üpris hästi dokumenteeritud ning lihtsasti kasutatav, ei olnud selle kaardielemendi seadistamine läbi Ignition Designer keskkonna lihtne, sest võimalused modifikatsioonideks olid vaid need, mida Ignition ise lubas. Kui algne plaan oli liikuda eemale markerite kasutamisest, sest nende hulk kaardil muutus probleemseks teatud piirkondades, kus on tihedam asustus, siis peale pikka üritamist ning ajaramis püsimiseks pidi ikkagi tegema valiku lihtsa, markertitega, lahenduse kasuks. Kaardilahenduse loomiseks oli vaja teada kõikide parkide koordinaate, mistõttu sai loodud iga pargi kohta metaandmete tabel. Tulevikuvaates sai loodud lahendus, et

metaandmete lisamine pargi juurde toimub juba pargi süsteemi lisamisel ning on määratud kohustuslikuks elemendiks.

Süsteemi kasutaja käsitsi töö vähendamiseks kirjutas lõputöö autor valmis ka skriptid, mis uuendasid võrrandeid, mille põhjal tehti arvutusi üle kõigi Sunly City klientide ning seadmete. Antud võrrandid võimaldavad kiiret ligipääsu andmetele ilma, et peaks minema ja tegema päringu iga pargi või seadme kohta eraldi kui varahaldajal on ees kõiki parke näitav monitoorimisvaade. Loodud skriptid jooksevad taimeriga määratud intervalli tagant ning toetavad süsteemi skaleerimist ning andmete korrektsuse aluseks olevate võrrandite püsivust ajas. Lisaks sellele, et enam ei pidanud neid võrrandeid käsitsi uuendada, mis on aeganõudev protsess, väheneb nii ka märgatavalt veategemise oht.

Üks põhifunktsionaalsusi, mis teeb moodsad SCADA platvormid väga mugavaks on nende sisseehitatud andmepäringute kirjutamiseks mõeldud tööriistad. Tavaline arendaja, kellel puudub programmeerimistaust, saab antud tööriistaga lihtsamad päringud tehtud, kuid kui süveneda antud tööriista dokumentatsiooni ning luua skriptid lisa andmetöötluseks muutuvad eelmainitud vahendid väga efektiivseks ning mugavaks. Seda funktsionaalsust sai ära kasutatud mitmes kohas. Esiteks leidis see kasutust klientide poolt defineeritavate ajavahemikes tunni- ja päevatootluse andmete pärimiseks (vt Lisa 8), mida kasutajad said Exceli formaadis alla tõmmata ning lisaks leidis see kasutust ka aasta-, kuu ja käesoleva päeva tootmisandmete päringute tegemiseks graafikute jaoks.

Rakenduse loomisel sai kasutatud lisandväärtuse loomiseks tasuta kättesaadavaid ilmaandmeid. Kuna ilmaandmed on hea element, mida kasutada päikeseparkide töö analüüsimiseks, siis oli igati mõistlik luua võimekus pärida hetkeilma andmeid läbi mõne rakendustarkvara liidese abil. Peale kiiret analüüsi, kus peamised kriteeriumid olid andmete edastamise formaat, tasuta päringute arv ning edastavate andmepunktide kogum sai välja valitud WeatherApi lahendus. Peale eelmainitud API-ga ühenduse loomist sai ilmaandmed ühendatud SCADA vaatesse nii, et uue pargi peale navigeerides tehti kohe päring ilmaandmete saamiseks ning kui kasutaja oli olnud sama pargi peal kümme minutit, siis toimus uuesti ilmaandmete uuendus.

Päikeseparkide kaugjälgimisel on üks olulisemaid komponente vaieldamatult alarmide ja muude probleemide tabamine. Selle jaoks sai käesoleva lahenduse puhul implementeeritud *backend*-poolel uus loogika alarmide ning

kommunikatsiooniprobleemide tuvastamiseks. Siin sai ära kasutada juba varem teise arendaja poolt loodud loogikat, et saata andmeid madalamalt tasemelt, seadmetelt, ülemistele tasemetele, parkide ja klientide andmeobjektideni, ning hiljem need kõik veel läbi võrrandite kokku liita kogu Sunly City alarme ning kommunikatsiooniprobleeme kokkuvõtivate andmeobjektide juures. Selline lähenemine kiirendas ning süstematiseeris veelgi andmete kättesaadavust monitoorimisvaate jaoks. Rakenduse kasutaja jaoks avaneb nupuvajutuse peale hüpinkaken, mis näitab problemaatilisi invertereid (vt Lisa 9). Järgnevalt liidestab lõputöö autor alarmide teavitused ka Slacki suhtlusplatvormiga, kus toimub Sunly firmasisene sõnumivahetus. Nimelt, Slack võimaldab saata grupivestlustesse e-maili teel teavitusi andes igale loodud kanalile unikaalse emaili aadressi. Nende e-mailide sisu kuvatakse kui tavalist sõnumit. Ignition Automation võimaldab seadistada e-maili kujul teavitusi läbi SMTP (lihtne meiliedastusprotokoll) serveri. Kasutades ära Ignitioni sisseehitatud alarmide käsitlemise moodulit Alarm Notification Pipelines sai loodud selline alarmist teavitav teavituskett (vt Lisa 10), mis saadab teavituse kui seadmega ei ole saadud ühendust ühe tunni ning hiljem nelja tunni järel. Sellise loogika kasutamiseks sai loodud *tag*, mis tabab ära hetke kui side katkeb või kui side seadmega on tagasi. Side katkemise järel hakkab tööle *tag*-ina loodud taimer, mis vastavalt tunni või nelja tunni täitudes paneb käima loodud teavitamisketi (vt Lisa 11). Olukorras kus side taastub lõpetab taimer ka sekundite lugemise.

## 5.2 Raporteerimislahenduse loomine

Raporteerimislahenduse loomisel sai ära kasutatud ühte Ignitioni tehnoloogia parimat omadust, milleks on erinevate lisamoodulite hilisem juurdeostmise võimalus. Raportite loomise jaoks on Ignition Automationil 3900 dollarit maksev moodul, mis võimaldab väga mugavalt genereerida tootmisraporteid. Kuivõrd kallist moodullahendust oli Sunly nõus finantseerima, sest seda nähakse kui investeringut ka tulevikus teistes projektides kasutamiseks. Kuna keegi Sunly SCADA-arendajatest ei olnud varasemalt antud lahendusega kokku puutunud, siis algas kogu protsess taaskord dokumentatsiooni lugemisest ning Inductive University õppevideote vaatamisest. Protsess raportite loomiseks oli selge, kõigepealt tuli valmis kirjutada andmepäringud, et saada andmed CSV (komaeraldusega väärtused) kujule. Selle jaoks sai ära kasutada juba varasema arenduse käigus kasutatust saanud lõputöö autori poolt loodud programmikoodi. Andmete visualiseerimiseks valiti erinevad Ignitioni poolt pakutavad graafikud.

Peale andmepäringute valmiskirjutamist on võimalik neid läbi programmikoodi veelkord vajadusel manipuleerida. Lisaks tootmisandmetele oli vaja raportisse ka ilmaandmed välja tuua ning siinjuures sai ära kasutada varasemalt loodud WeatherApi ühendust.

Raporteerimislahenduse puhul oli oluline disain ning sõnastuse korrektsus. Ajalise piirangu tõttu oleks need aga olnud kaks lisäülesannet, mida ei oleks võib-olla jõudnud arenduse ajaplaani ära mahutada. Õnneks on töötab Sunly-s graafiline disainer ning isik, kes kontrollib tekstide firma keelekasutusega kokkulangevust, mistõttu õnnestus antud ülesanded nende abil kiirelt ära teha. Raporti disainimise kohapealt sai lõputöö autor abi läbi selle, et graafiline disainer tegi valmis SVG (skaleeruv vektorgraafika) kujul raportipõhja ka varasemalt kasutatust leidnud keskkonnas Figma. Soovitav disain sai kooskõlastatud koosolekuga. Ignitioni raporteerimismoodulis on funktsionaalsus, mis lubab importida raportipõhja disaini SVG kujul ning siis sinna peale informatsiooni paigutada. Näide raportipõhja esimesest lehest leiab Lisa 12 alt. Raportipõhja peale saab informatsiooni paigutada järgnevalt kirjeldatava loogika alusel. Esiteks tuleb tuua sobib element, näiteks tekstikast, raportipõhja peale ning määrata ära selle suurus ning muud atribuudid. Peale seda tuleb sellele elemendile anda nimi, mille alusel saab raportit genereerides läbi programmikoodi erinevatele elementidele väärtusi määrata. Kirjeldatud funktsionaalsusega oli väga lihtne peale vastava Inductive University koolituse läbimist tööd teha. Sarnaselt SVG formaadis raportipõhjale, saab raporti loomisel kasutada ka PDF (porditav failivorming) põhja.

Eraldi väljatoomist väärrib veel kaotatud toodangu arvutus. Kaotatud toodangu arvutus peab näitama kliendile, kui suur hulk energiat jäi tootmata lahenduse pakkuja-poolse probleemi tõttu. Kaotatud toodangu arvutus ei kohaldu seega näiteks võrguprobleemide korral. Esimene lahenduskäik selle arvutuse jaoks põhineks tolle hetke päikesekiirguse intensiivsusel. Kahjuks aga nii väikestesse parkidesse ei paigaldata päikesekiirgus sensoreid finantsilistel põhjustel ning seetõttu tuli leida alternatiivne viis kaotatud toodangu arvutuse tegemiseks. Väljamõeldud lahendus põhines lähedalasuvate parkide tootmisandmetele. Lõputöö autor tegi valmis loogika, mis leiab igale pargile kolm kõige lähemal asuvat Sunly päikeseparki ning arvutab nendelt saadava informatsiooni põhjal kui palju jäi mingil rikkeperioodil energiat tootmata. Et aga selline arvutus toimiks, tuli hakata iga päikesepargi kohta koguma süsteemi metaandmeid, mis tähendas, et tagantjärele tuli lisada kõikidele päikeseparkidele nende kWp (tipuvõimsus kilovattides) ning inverteri võimsus. Lisaks tuli siinkohal lõputöö autoril luua loogika, et

asendusarvutuseks võis kasutada vaid parki, kus ei ole hetkel aktiivset alarmi, sest muidu tehakse arvutusi ebasobivate andmete pealt. Just eelmainitud põhjusel saigi valitud kolm kõige lähemal asuvat parki, et nii kaitsta loogikat selle eest, et mitmes pargis võib mingil põhjusel korraga probleeme olla. Kolm kõige lähemal asuvat parki arvutatakse ümber igakord kui süsteemi lisatakse või süsteemist eemaldatakse päikesepark. Kuna toodangut logitakse kümneminutilise intervalli tagant, siis oli mõistlik ka kaotatud toodangu arvutust rakendada kümneminutilise intervalliga. Iga lisanduva pargiga muutub arvutus täpsemaks, sest vahemaad lähedal asuvate parkide vahel muutuvad väiksemaks.

Lõputöö varasemas osas, kus lõputöö autor analüüsis erinevate monitoorimisrakenduste omadusi, sai välja toodud, et monitoorimisrakenduse vaatenurgast on õhku paiskamata jäänud süsinikdioksiidi hulk pigem ebavajalik. Raporteerimislahenduse puhul, mis on mõeldud kliendile, võib aga antud näitajat ära kasutada küll ning mõnele kliendile pakuks see kindlasti huvi. Antud informatsiooni raportisse panemiseks uuris lõputöö autor, kuidas eelmainitud arvutust tehakse. Arvutuse tegemiseks oli vaja ainult kahte väärtust, milleks olid toodangu hulk ning emissioonifaktor.

Raportite jaoks tuli valmis teha kolm erinevat disaini, et katta ära kolm võimalikku stsenaariumi. Need kolm stsenaariumi olid klient ühe pargi ja ühe inverteriga, klient ühe pargi ja mitme inverteriga ning klient mitme pargi ja mitme inverteriga. Kuigi suur osa raportitesse lisatavatest andmetest olid sarnased, siis oli elemente, näiteks graafikud, mille puhul tekkisid erinevused. Siinkohal tuleb välja Ignition raporteerimismooduli üks märkimisväärseid nõrkusi ehk et see ei ole väga dünaamiline.

Raporti väljasaatmine toimub e-maili abil. Raportite väljasaatmiseks sai loodud juba varem eksisteerinud SMTP serverisse uus meilikonto. Raportite väljasaatmiseks on kaks võimalust. Esimene neist on käsitsi ning seda saab teha mõni isik Sunly poolelt läbi monitoorimisrakenduse omades selleks operatsiooniks vajaminevaid õigusi. Teine võimalus raportite väljasaatmiseks on automatiseeritud ning kasutab ära taaskord Ignitioni *tag*-ide funktsionaalsust. Seekord kasutas lõputöö autor *tag*-i aga ajastaja loomiseks. Nimelt on üks *tagi*-i tüüp, mille sisemine loogika aktiveerub kasutaja defineeritud võrrandi alusel. Kombineerides seda võimekust võimalusega pärida hetke ajatemplit lõi lõputöö autor sellise võrrandi, mis uuendab *tag*-i väärtust vastavalt sellele, mis kuupäev antud ajatemplitil on. Peale eelmainitud *tag*-i loomist sai loodud loogika, mis tuvastab kui kuupäeva arvutav *tag* muudab oma väärtust ehk kui tuleb uus päev. Kuu lõpp

tuvastatakse, kui uus võrrandi poolt arvutatud väärtus on väiksem kui eelmine väljaarvutatud väärtus. Selline olukord tekib, kui kuupäev liigub näiteks 31.jaanuarilt 1.veebruari peale. Eelmainitud võrrand uuendab ennast iga minuti tagant ehk kuuvahetuse tuvastab süsteem väga kiirelt. Peale kuuvahetuse tuvastamist läheb käima loogika, mis kutsub välja funktsiooni, mis loob ja saadab laiali kuuraportid. Kahjuks aga ei saa teist automatiseeritud lahendust lõputöö kirjutamisel kasutada, sest lõputöö autor ei leidnud vajaminevate andmete pärimiseks mõistliku hinnaga rakendusliidest.

### 5.3 Testimine ning tagasiside

Vastavalt uue tarkvara testimise põhitõdedele tuleks iga vaade hoolikalt testida enne selle rakendamist reaalses keskkonnas. Antud lõputöö puhul tähendas see ka olemasolevate arendusprotsesside ümber tegemist. Enne lõputöö autori poolt loodud lahendust toimusid muudatused otse lõpliku lahenduse keskkonnas. See tähendas, et kui programmikoodis oli kuskil mingisugune kriitiline viga, siis võis see kasutuses oleva süsteemi tõsiste tagajärgedega maha tõmmata. Seetõttu võttis lõputöö autor kasutusele ühe sülearvuti, kuhu sai installeerida Ignition Gateway serveri. Kuna Ignition pakub tasuta prooviversiooni, kus on kogu funktsionaalsus avatud, siis sai sellest ideaalne keskkond uute lahenduste testimiseks. Eelmainitud lahenduse ainukeseks probleemiks oli see, et iga kahe tunni tagant peab manuaalselt vajutama prooviversioonis *reset* nuppu. Selline lähenemine testimisele on soovitatuna välja toodud ka Inductive Automationi ametlikus dokumentatsioonis parimate arenduspraktikate all.

Testimise esimene samm oli tagada, et kõik seadmed ja protsessid oleksid korrektselt integreeritud süsteemi ja kajastatud loodud rakenduses. Kuna suhtlus seadmete ja rakenduse vahel toimib varasemalt mainitud OPC UA ning Modbus TCP protokollide läbi siis oli oluline kontrollida läbitulevate andmete korrektsust. Antud protsess oli üpris lihtne tänu Ignitioni sisseehitatud võimekusele tuvastada andmeid, mis on eelnevalt seadistatud piirangutest väljas. Taoline võimekus oli järjekordne märk sellest, et lõputöö tehnoloogia valikul sai tehtud õige otsus.

Järgnevalt oli oluline kontrollida, kuidas vaade toimib erinevates seadmetes nagu näiteks telekas ja arvutimonitor. Erienvatel ekraanidel testimise käigus tuli välja, et vaade ei skaleeru kõige paremini väikeste sülearvutimonitoride peal aga telekate ning lauaarvutite jaoks mõeldud monitoride peal toimis vaade väga hästi. Kuid kuna antud töö puhul oli

aktsepteerivaks kriteeriumiks see, et see töötab telekatel ning keskmisest suurematel monitoridel, siis jäi antud kitsaskoha eemaldamine hetkel kõrvalplaanile. Hilisema platvormi uurimise tulemusena tuli välja, et antud probleemi oleks olnud tegelikult lihtne lahendada kui lõputöö autor oleks kasutanud erinevat baasdisaini komponenti, mida Ignition pakub.

Edasi tuli teostada kasutatavustestid erinevatele kasutajagruppidele. See võimaldas kindlaks teha, kas vaade vastab nende erinevatele vajadustele ja ootustele. Kasutajagruppidele sai antud ligipääs loodud rakendusele ning sai lahti seletatud, mida mingi element või funktsionaalsus endast kujutab. Tagasiside käigus tuli välja, et rakendus on mugav, kuid siiski võiks selle jaoks eksisteerida mingisugune kasutajajuhend, mida välja saata. Lõputöö autor ootas isiklikult pisut rohkem kriitikat rakenduse väljanägemise kohta, kuid sellekohane negatiivne tagasiside oli väga minimaalne.

Lisaks peaks testimisel arvestama ka erinevate võimalike stsenaariumitega, nagu seadme rike või päikesepaneelide madal tootlikkus, ja kontrollima, kuidas vaade neid kuvab. Selliste olukordade tekitamiseks sai loodud eriline pargi objekt, mida sai kontrollida nii, et see genereeriks tootmisandmetele sarnaseid näitajaid ning tekitaks vajadusel alarme ja veateateid.

Kuigi rakenduse ning selle erinevate funktsionaalsuste jaoks kasutatakse päris palju koodi on selle testimine üpris keeruline, sest ühtegi mugavat testimisteedi või -moodulit Ignition ei paku. See tähendas, et koodi testimiseks pidi lõputöö autor kasutama lihtsalt konsooli ning seal erinevaid meetodeid erinevate potentsiaalselt problemaatiliste parameetritega välja kutsuma. Selline lähenemine ei ole aga kindlasti pikaajaliselt jätkusuutlik ning kui projektiga oleks tegelenud mitu arendajat oleks see olnud suureks kitsaskohaks.

## **5.4 Tagasiside põhjal tehtud täiendused**

Kuigi tagasiside, mida erinevad kasutajagrupid andsid oli enamasti positiivne, siis oli ka kohti, mida toodi välja kui puudulikud. Nimelt, Sunly Cityl on kliente kellel on palju parke ning, kes sooviksid andmeanalüüsiks saada andmeid nii, et nad saaksid need oma andmelattu paigutada. Algsel nõuete kaardistamisel oli see osa välja jäänud. Lisaks oli nende jaoks kitsaskohaks ka see, et loodud lahenduse poolt pakutavad allatõmmatavad



tootmisandmed ei olnud nende jaoks sobival ehk masinale lihtsasti töödeldaval kujul vaid olid loodud inimloetavaks. Antud probleemi tõstatanud kliendi puhul tähendas see CSV formaadis tootmisandmete saatmist SFTP (turvaline failiedastus protokoll) serverisse. Kliendiga peetud koosoleku tulemusena sai paika pandud vajaminevate andmete, andmete struktuuri ning ligipääsu informatsioon. Vajaminevate päringute kirjutamine oli lihtne, sest sai taaskasutada projekti varasemas faasis loodud funktsionaalsust. Lisandus ka paar lisanõuet, millest vast kõige märkimisväärsem oli EIC-koodide olemasolu andmetes. See tähendas, et lõputöö autor pidi looma eraldi metaandmete tabeli, kus sai hakata hoidma iga päikesepargiga kokkuminevat lisainfot. Kui andmepäringud töötasid, siis edasi oli vaja uurida kuidas oleks kõige mõistlikum lahendada seda, et neid andmeid ajastusega SFTP serverisse saata. Taaskord oli abiks Ignitioni sisseehitatud funktsionaalsused, mida sai lõputöö autor enda jaoks ära kasutada. Ajastamine on Ignitioni tarkvaraga lihtne tänu võrrand tüüpi *tag*-idele, mis võimaldavad tabada ära kuuvahetuse. Nii sai loodud *tag*, mis uue kuu teisel päeval kogub kokku eelneva kuu tootmisandmed. Selleks kasutas lõputöö autor Jythoni programmikoodi ning Ignitionisse sisseehitatud teeki ajaloolist infot salvestavate *tag*-ide pihta päringute tegemiseks. Peale päringute tegemist sai viidud andmed sobivale CSV kujule. Järgmine samm oli CSV saatmine Hetzneri pilveserveris jooksvale konteinerlahendusele, mille tegi lõputöö autor selleks, et oleks võimalik luua lahendus SFTP serverisse saatmiseks Python keeles. Lõputöö autor tegi sellise otsuse, sest Python programmeerimiskeelel on rohkem teeki, mida on mugav kasutada SFTP serverisse andmete saatmiseks ning tulevikus ka teiste erilahenduste jaoks. Konteinerlahendusest edasi saadetakse CSV kliendi poolt antud kasutajanime, parooli, dokumendikausta ning aadressi alusel kliendi SFTP serverisse.



Joonis 2. Kliendi tootmisandmete SFTP serverisse saatmine CSV kujul

Lisaks eelmaintule sai tehtud ka väiksemaid korrekture. Näiteks sai uuendatud teksti, mida kuvatakse kui klikkida problemaatilisi invertereid kajastava nupu peale. Algselt toodi sinna välja väga täpne veakood aga tagasiside põhjal sai antud informatsioon lihtsamaks ning üldisemaks. Varahaldaja seisukohast ei ole selline lahendus muidugi ideaalne aga kuna Ignition võimaldab defineerida kasutajarolle, siis sai seda antud

olukorras mugavalt ära kasutada ning luua lahenduse, kus kliendile ning varahaldurile kuvatakse erinevat informatsiooni.

## 6 Hinnang rakendatud uuendustele

Võrreldes algselt eksisteerinud lahendusega on edasimineku lõputöö autori arvates olnud märkimisväärne. Arendajapoolne käsitsi tehtav töö uue päikeseparki süsteemi integreerimisel on oluliselt vähenenud ning andmepäringud on muudetud rohkem süstematiseerituks ja arusaadavaks. Graafilise poole pealt on lõputöö autor lahendusega rahul, kuigi sisseehitatud CSS funktsionaalsust sai vähe kasutatud. Seda saab aga põhjendada läbi selle, et Ignitioni enda sisseehitatud komponentide kujundamise muutmiseks kasutatavad tööriistad oli antud lahenduse kontekstis piisavad ning üllatavalt kasutajasõbralikud isegi keerulisemate elementide puhul.

Varahaldusvaatest saadava informatsiooni hulk oli erinevate kasutajagruppide hinnangul hea ehk see andis märku sellest, et nõuete kaardistamise protsess oli olnud edukas. Kuigi analüüsi käigus vaatluse all olnud konkureerivad lahendused näitasid rohkem informatsiooni ei leidnud rakendust testinud ning nüüdseks igapäevaselt kasutavad inimesed, et neil oleks milleski puudu.

Klientidele tootmisandmete kättesaadavaks tegemine Exceli tabelina allalaadimise ning teatud klientide puhul CSV failide SFTP serverisse saatmise näol on olnud nii Sunly City kui ka klientide sõnul suur edasimineku ning palju lisaväärtust pakkuv lahendus. Antud funktsionaalsuse puhul tuleb aga arvesse võtta, et tulevikus võib tekkida olukord, kus on palju kliente, kes kõik soovivad andmeid vaid neile sobival kujul ning see võib tähendada, et rakenduse programmikood, mis pärib ning vormistab andmeid sobivale kujule muutub erandite tõttu raskesti hallatavaks.

Andmete hoiustamise ja päringute kiirendamise vaatenurgast oleks olnud mõistlik luua lisaks kronoloogiliselt logitava info andmebaasile selline andmebaas, kus on juba kokkuvõtlikud andmed valmis arvutatud. Kuigi Ignitioni tööriistad töötavad kronoloogilistelt salvestatud andmebaasidega väga efektiivselt, oleks siiski olnud mõistlik teha tabelid kokkuvõtivate andmete jaoks, sest näiteks kogu Sunly City kuu tootmisandmete päring võtab praeguse lahenduse puhul paar sekundit aga kokkuvõtivate tabelite abiga oleks andmete pärimiseks kuluv aeg märkamatu. Antud probleem tuleb ilmsiks ainult kõiki Sunly City päikeseparke monitoorivas vaates, kus on lõputöö kirjutamise hetkel 61 päikeseparki. Siiski, siinkohal on suur oht, et antud lahendus hakkab mõjutama ka kliente, kellel on rohkem kui paar päikeseparki. Selline eraldi kokkuvõtvaid andmeid hoidev tabel oleks tõenäoliselt kiirendanud ka raportite jaoks tehtavate päringute

aega aga kuna hetkel võtab see protsess suurima kliendi puhul siiski vaid mõne sekundi siis ei tohiks see probleemiks olla.

Raporteerimislahenduse puhul saab lõputöö autori arvates olla rahul loodud lahendusega vaid osaliselt. Kuigi raport annab hea ülevaate päikesepargi tööst on selle väljasaatmine tülikas, mistõttu on tõenäoline, et see tuleb tulevikus ümber teha. Nimelt, raporteerimislahenduse puhul on probleemiks päikeseinfo saamine. See oli aga üks osa raportist, mis ühe osapoole arvates pidi seal kindlasti olema, mistõttu tuli luua parim võimalik lahendus. Probleemiks on, et päikeseinfo pärimine läbi mõne API on kallis. Lõputöö autor otsis alternatiive probleemi lahendamiseks ning parimaks alternatiiviks osutus päikesetundide manuaalne süsteemi sisestamine. Kuigi tegemist on lõputöö autori hinnangul parima võimaliku alternatiiviga, on lõputöö autori arvates protsess ikkagi piisavalt ebamugav ning aeganõudev, et selle kasutamine ei ole pikaajaliselt jätkusuutlik. Äramärkimist väärib fakt, et loogika täisautomatiseeritud lahenduse jaoks on olemas, kui selle jaoks peaks kunagi raha tekkima.

Lõputöö autori arvates on loodud lahendus kinni pidanud eelnevalt väljatoodud neljast kasutajakogemuse printsiibist. Loodud vaade ei ole liiga kirju, ei sisalda ebavajalikku ega ole keeruline tavakasutaja jaoks.

## 7 Ettepanekud edasiseks arenduseks

Antud lõputöö käigus loodud rakenduse puhul on ideid tulevaseks arenduseks küllaga ning suur osa väljatoodavatest ideedest näevad tõenäoliselt lähiajal ilmavalgust. Kuna tegemist oli üpriski uue arenduskeskkonnaga, siis sai lõputöö autor rakendust luues mitmeid õppetunde, mida edasise arenduse käigus saab ära kasutada efektiivsemaks rakenduse loomiseks ning parema lahenduse loomiseks.

Lõputöö autori hinnangul väärib äramärkimist kaardikomponent, mis kajastab parkide asukohti. Lõppkasutajad on loodud lahendusega rahul, kuid lõputöö autor näeb, et võib tekkida probleeme, mida sisseehitatud kaardikomponendiga lahendada ei saa. Selline probleem võib tekkida, kui päikeseparkide arv muutub nii suureks, et markerid hakkavad üksteist katma. See tähendab, et informatsioon, mida markerid peaksid edasi andma, läheb kaotsi. Selline olukord on Eesti populatsiooni paiknemist arvesse võttes täiesti reaalne probleem, sest suur osa Eesti elanikkonnast elab Tallinnas ning Tallinna lähiümbruses. Antud probleemi oleks võimalik ära lahendada näiteks soojuskaardi abil. Soojuskaardi funktsionaalsust ei ole Ignition Automation ise veel välja arendanud aga seda on võimalik kätte saada läbi kolmanda osapoole moodulite.

Edasises arenduses on suur roll nii elektrihinna järgi juhtimissüsteemi loomisel kui ka energiasalvestuse juhtimine vastavalt päiksepargi toodangule. Juhtimise peamine eesmärk oleks aidata klientidel oma elu korraldada vastavalt olemasolevale soodsale energiale ning mitte Nordpool elektriturule hinnale, mis on mõjutatud palju tehnilisest oludest nagu näiteks Estlinki ja Auvere elektrijaama tehnilised rikked.

Kujundus oli antud projekti puhul teisejärguline, kuid kui mõelda klientide kogemuse peale siis oleks visuaalselt meeldivam lahendus kindlasti midagi sellist, mille poole peaks edasiste arenduste käigus püüdlema. Ignition platvormil on Perspective moodulis väga hea tugi erienvate laadilehtede loomiseks. Sarnaselt sellele, kuidas Ignitionis on võimalik hoida programmikoodi selleks eraldi ettenähtud kohas on seda võimalik teha ka CSS faili puhul.

Võrreldes loodud rakendusega võiks edasiste arenduste puhul kindlasti rohkem ära kasutada ka Ignitionisse sisseehitatud kasutajarollide süsteemi. Kui hetkel on lõppkasutajad jaotatud kaheks, kliendid ja varahaldurid, siis tulevikus võiks siinkohal

diferentseerida lisaks äri -ja erakliendi vahel. Lisarollide loomine võimaldaks pakkuda kindlasti palju personaalsemaid lahendusi ning pakub võimalusi tulevikus näiteks rakenduse tasuta versiooni loomiseks.

Efektivsusele ning skaleeritavusele aitaks kaasa ka varasemalt (peatükk 6) väljatoodud ajaloolisi andmeid kokkuvõtivate andmetabelite idee. Selline edasiarendus oleks tulevikus kindlasti vajalik, sest muidu muutub olukorras, kus ühe kliendi päikeseparkide arv on väga suur raportite genereerimise aeg ebamugavalt pikaks.

Ideid edasiseks arenduseks hakkab tulevikus kindlasti tulema ka kasutajatelt endilt. Antud lahenduse puhul ongi klientide jaoks üks kindel eelis see, et kuna kliendibaas ei ole tuhandeid kliente siis on võimalik arendajatel erisoovidele vastu tulla ning nende ideed ära kuulata. Üks selliseid ettepanekuid on olnud näiteks, et teavitused pargi probleemidest võiks tulla telefonile sõnumina, sest tavakliendid ei hoida monitoorimisrakendust pidevalt taustal. Selline edasiarendus oleks lõputöö autori arvates vägagi väärtuslik klientide puhul, kes sooviksid olla ise rohkem seotud oma päikesepargi parimas seisus hoidmisega.

## 8 Kokkuvõte

Antud lõputööga loodi varahalduse ja aruandluse lahendus päikeseparkide pakkujale ja selle klientidele, mis aitab varahalduritel, operaatoritel ja projektijuhtidel teha päikeseparkide optimaalse töö tagamiseks informeeritud ning kiireid otsuseid.

Lõputöö autor pidas erinevate osapooltega koosolekuid, mille tulemuseks olid kasutajalood, mis olid aluseks rakenduse nõuete defineerimisel. Töö käigus analüüsis autor hetkel olemasoleva lahenduse puudujääke ning teisi Sunly SCADA meeskonna poolt tehtud projekte, tuues välja, mida saaks käesoleva lahenduse puhul parandada. Analüüsi käigus uuris lõputöö autor teiste päikeseparkide monitoorimiseks kasutatavate lahenduste kohta. Võrdluse ning analüüsi tulemusena tõi lõputöö autor välja erinevate lahenduste tugevusi ning nõrkusi. Tehnoloogia valiku põhjendamisel tõi autor välja põhjused, miks ta antud lõputöö kontekstis kasutas traditsiooniliste veebiarendustehnikate asemel modernsete SCADA lahenduste pakutavaid rakenduste arendamiseks mõeldud tööriistu.

Lõputöö käigus saadud kogemuse põhjal on SCADA platvorme, mis lõputöö autori arvates on piisavalt head, et asendada tavalisi veebiarendustehnikaid paljudes, kuid mitte kõikides olukordades. Lõputöö projektina loodud lahendus oli kindlasti üks nendest olukordadest, kus arendajal on võimalus ära teha samaväärne töö SCADA platvormiga, eriti olukorras, kus ajaline piirang mängib suurt rolli.

Valminud rakendus lahendab aktuaalset probleemi ning pakub võimalusi edasiarenduseks, mis optimeeriks veelgi päikesepargi monitoorimist ning aitaksid kaasa taastuvenergiale üleminekul.

## Kasutatud kirjanduse loetelu

- [1] Masterson, V. (2021, July 5). Renewables Were The World's Cheapest Energy Source in 2020. World Economic Forum. Retrieved November 17, 2023, from <https://www.weforum.org/agenda/2021/07/renewables-cheapest-energy-source/>
- [2] Eservice. (n.d.). Pääkese elektriijaamad. Retrieved December 1, 2023, from <https://eservice.ee/paikeseelektriijaamad/>
- [3] Abdulsalam, A. B., Alsaadi, H. A. J., & Hamodat, Z. (2022, June 9). Control and management of solar PV grid using Scada system. 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). Presented at the 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey. doi:10.1109/hora55278.2022.9799994
- [4] Inductive Automation. (n.d.). Inductive Automation documentation. Retrieved September 1, 2023, from <https://docs.inductiveautomation.com/>
- [5] Inductive Automation. (n.d.). Ignition 8 Deployment Best Practices. Retrieved November 11, 2023, from <https://docs.inductiveautomation.com/display/DOC81/Ignition+8+Deployment+Best+Practices>
- [6] Cimlogic. (2023, June 19). The Ignition Advantage: A Modern Approach to SCADA Systems. Retrieved November 12, 2023, from <https://www.cimlogic.co.uk/blog/the-ignition-advantage-a-modern-approach-to-scada-systems/>
- [7] Hamus, C. (2017, November 28). Web Development Technology for SCADA Systems. Retrieved November 12, 2023, from <https://www.vertech.com/blog/web-development-technology-for-scada-systems>
- [8] Balsom, P. (2019, September 30). The Role of Programmable Logic Controllers in a SCADA System. High Tide Technologies. Retrieved October 22, 2023, from <https://htt.io/the-role-of-programmable-logic-controllers-in-a-scada-system/>
- [9] Levanduski, M. (2023, June 14). Ignition SCADA Software: What are Tags and How are They Used?. Retrieved October 7, 2023, from <https://control.com/technical-articles/ignition-scada-software-what-are-tags-and-how-are-they-used/>
- [10] Tariq, H. (2021, May 7 ). Low-Code Versus No-Code And The Future Of Application Development. Forbes. Retrieved November 5, 2023, from <https://www.forbes.com/sites/forbescommunicationscouncil/2021/05/07/low-code-versus-no-code-and-the-future-of-application-development>
- [11] Siggins, M. (2020, October 13). Top 10 Features That Modern SCADA Systems Have. Retrieved October 7, 2023, from <https://www.dpstele.com/blog/10-features-modern-scada-systems-have.php>
- [12] AVEVA. (n.d.). System Platform. Retrieved October 7, 2023, from <https://www.aveva.com/en/products/system-platform/>
- [13] Energy XPRT. (n.d.). PowerOn - Reliance Energy Management System Software. Retrieved October 7, 2023, from <https://www.energy-xprt.com/software/poweron-reliance-energy-management-system-software-323172>
- [14] Gartner. (n.d.). GE PowerOn ADMS Reviews, Ratings & Features 2023. Retrieved October 7, 2023, from <https://www.gartner.com/reviews/market/advanced-distribution-management-systems/vendor/ge/product/ge-poweron-adms>



[15] Krug, S. (2014). Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability (3. version). New Riders.

## Lisa 1. Lihtlitsents

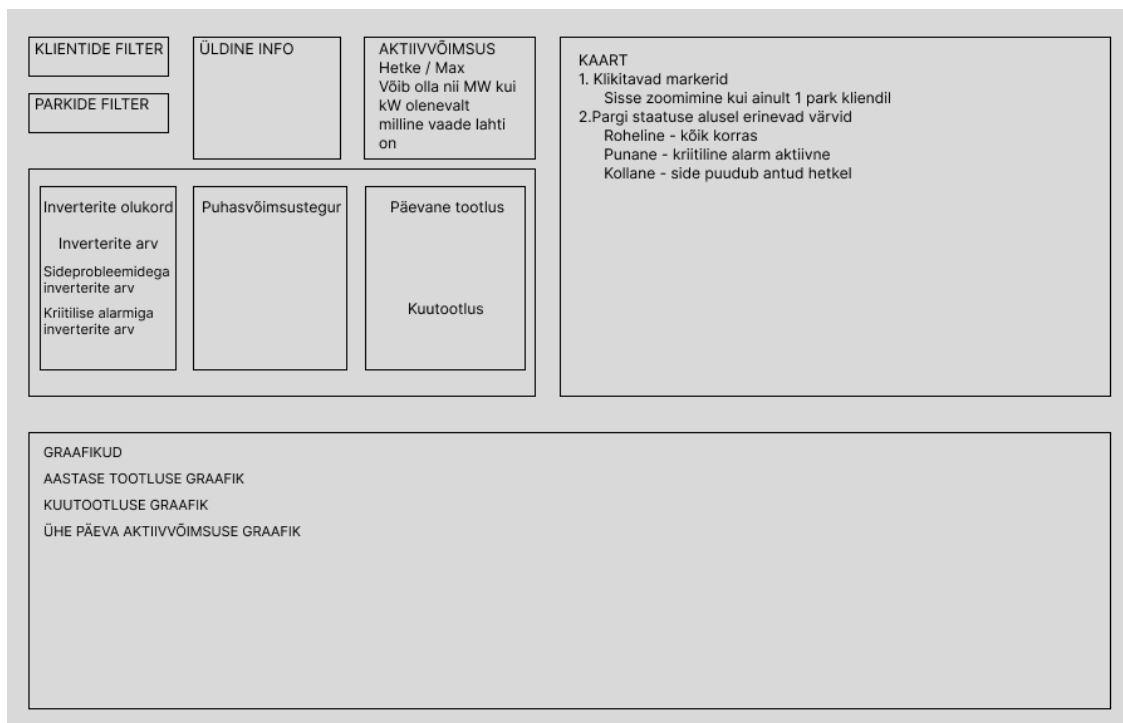
Mina, Martin Siitam

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud lõputöö "Varahalduse ja aruandluse lahendus päikeseparkide pakkujale ja selle klientidele", mille juhendaja on Meelis Antoi ning kaasjuhendaja on Taavi Sarnet
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

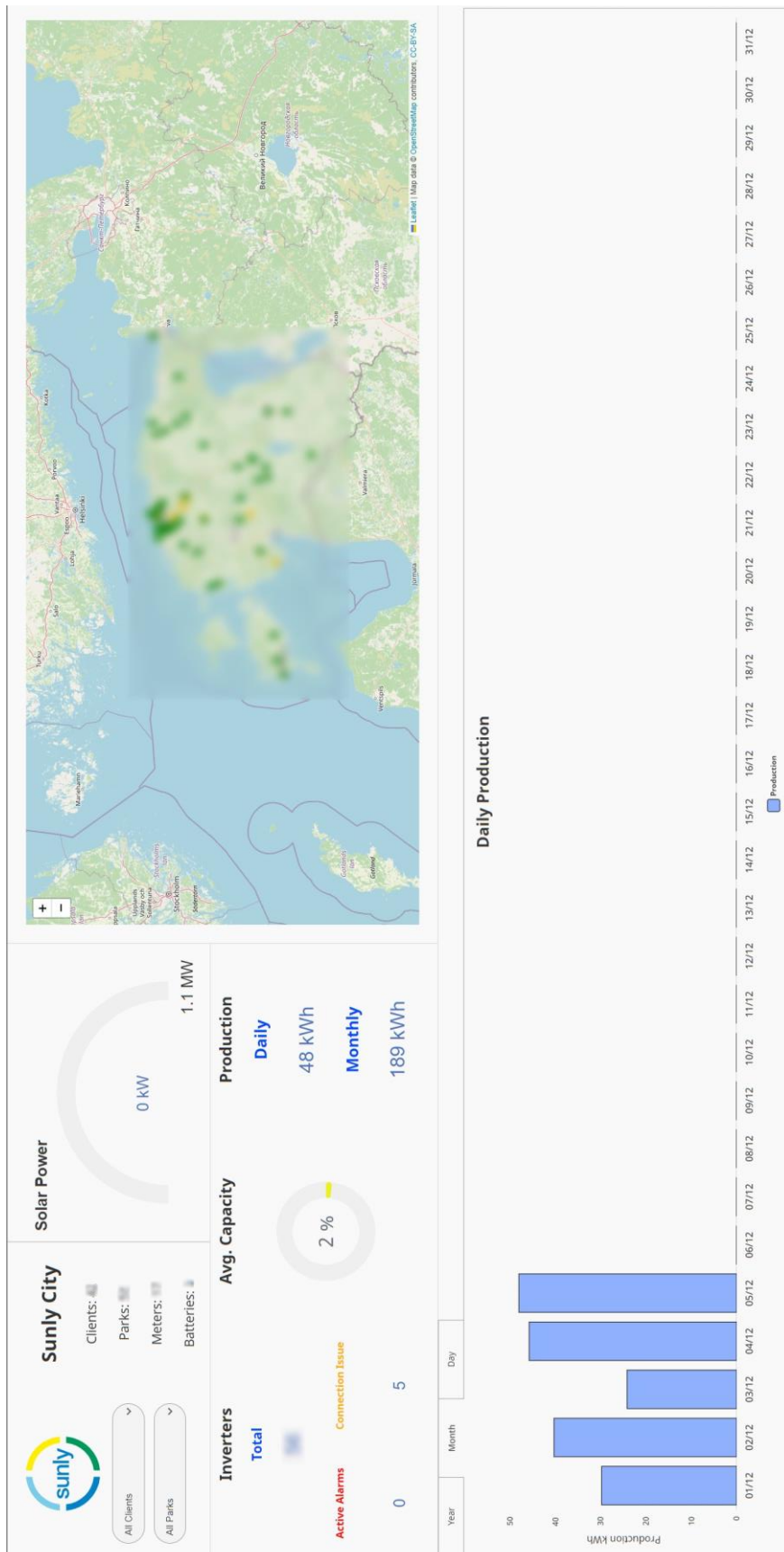
# Lisa 2. Programmikoodi kirjutamine Ignition Designer arenduskeskkonnas



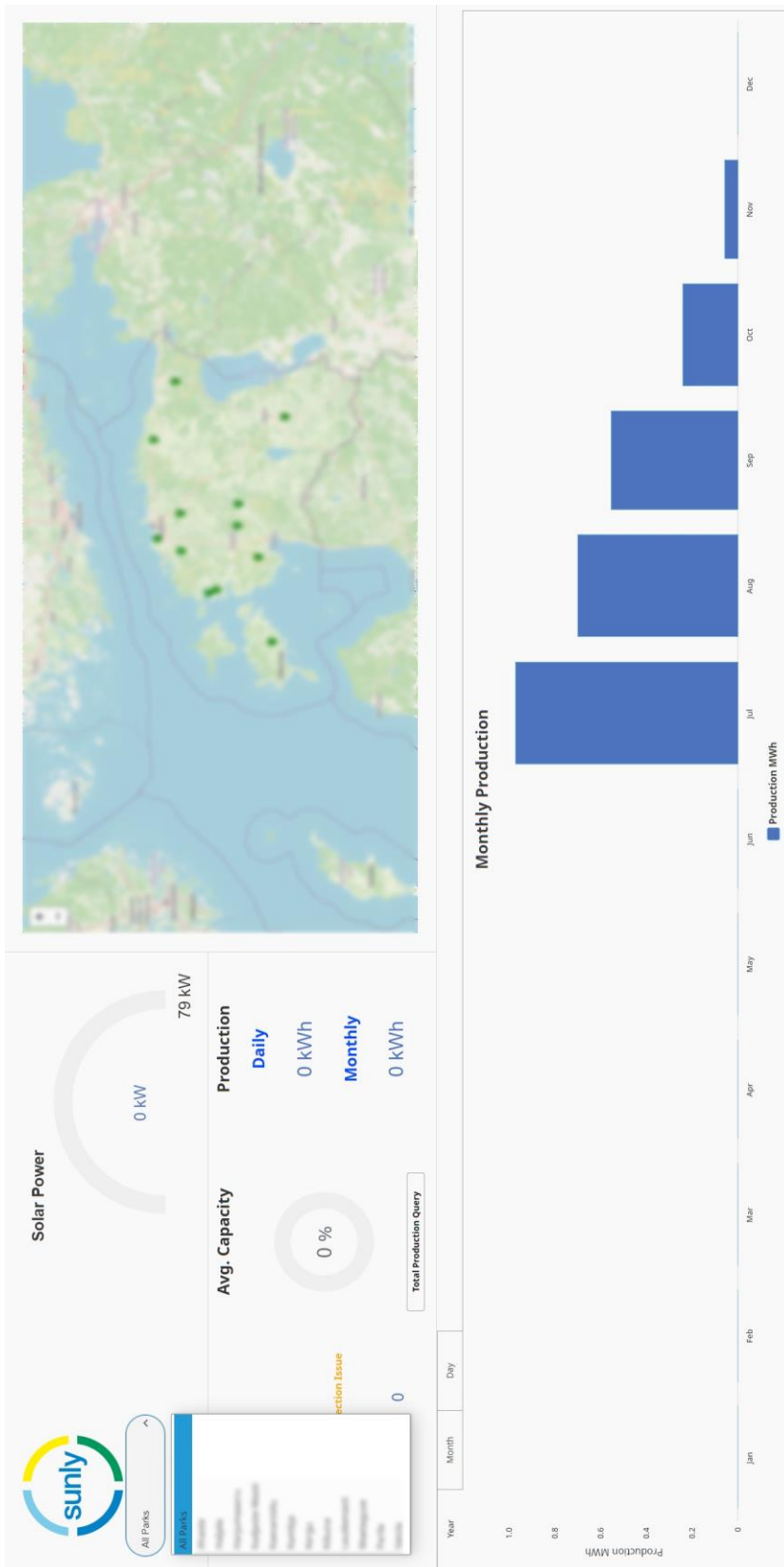
## Lisa 3. Visand monitoorimisrakenduse vaatest Figma keskkonnas



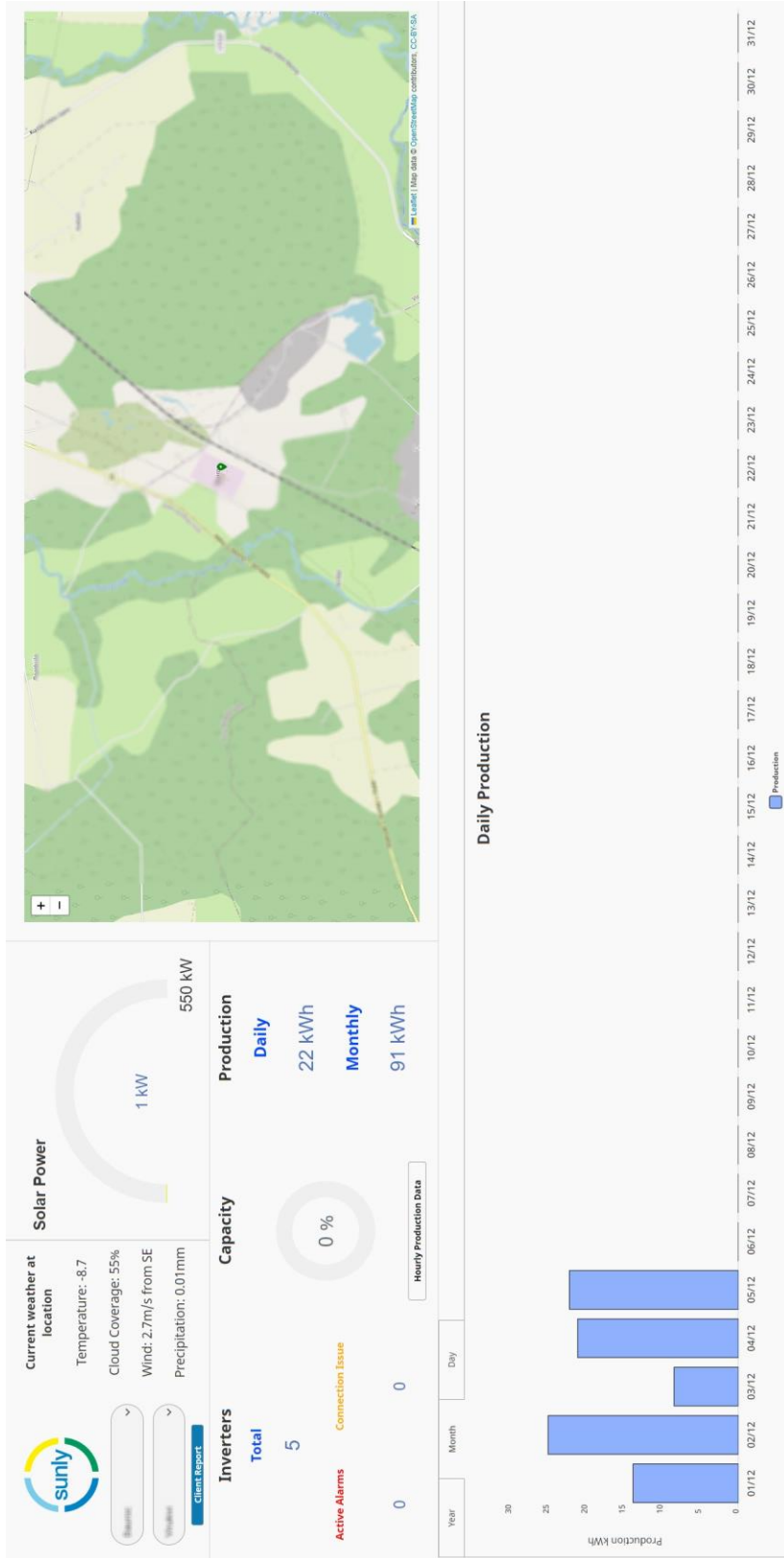
# Lisa 4. Varahaldurile avanev vaade kõikide päikeseparkidega



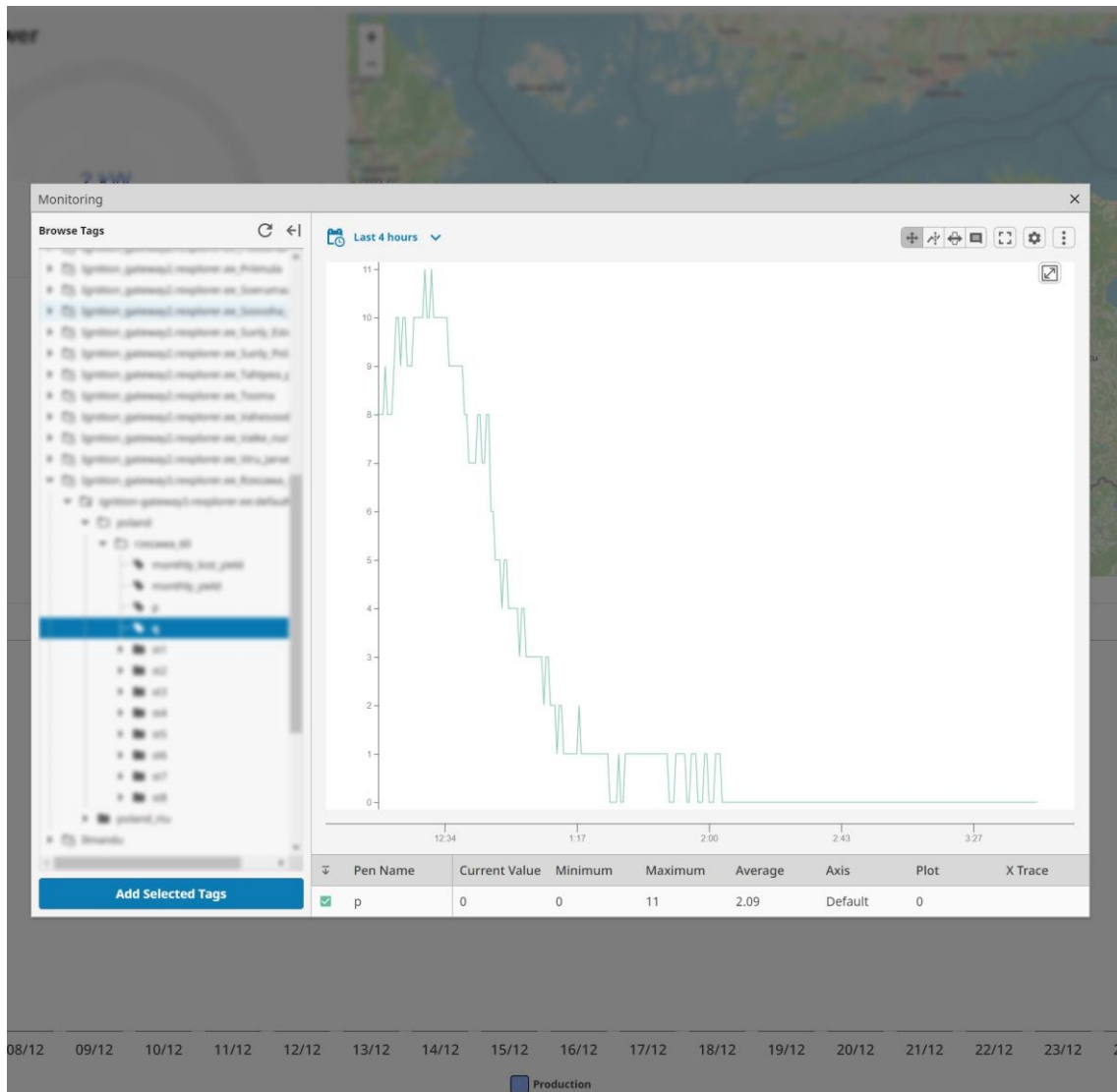
# Lisa 5. Kliendi vaade kliendi kasutajaõigustega



# Lisa 6. Varahalduri kasutajaõigustega kliendivaade

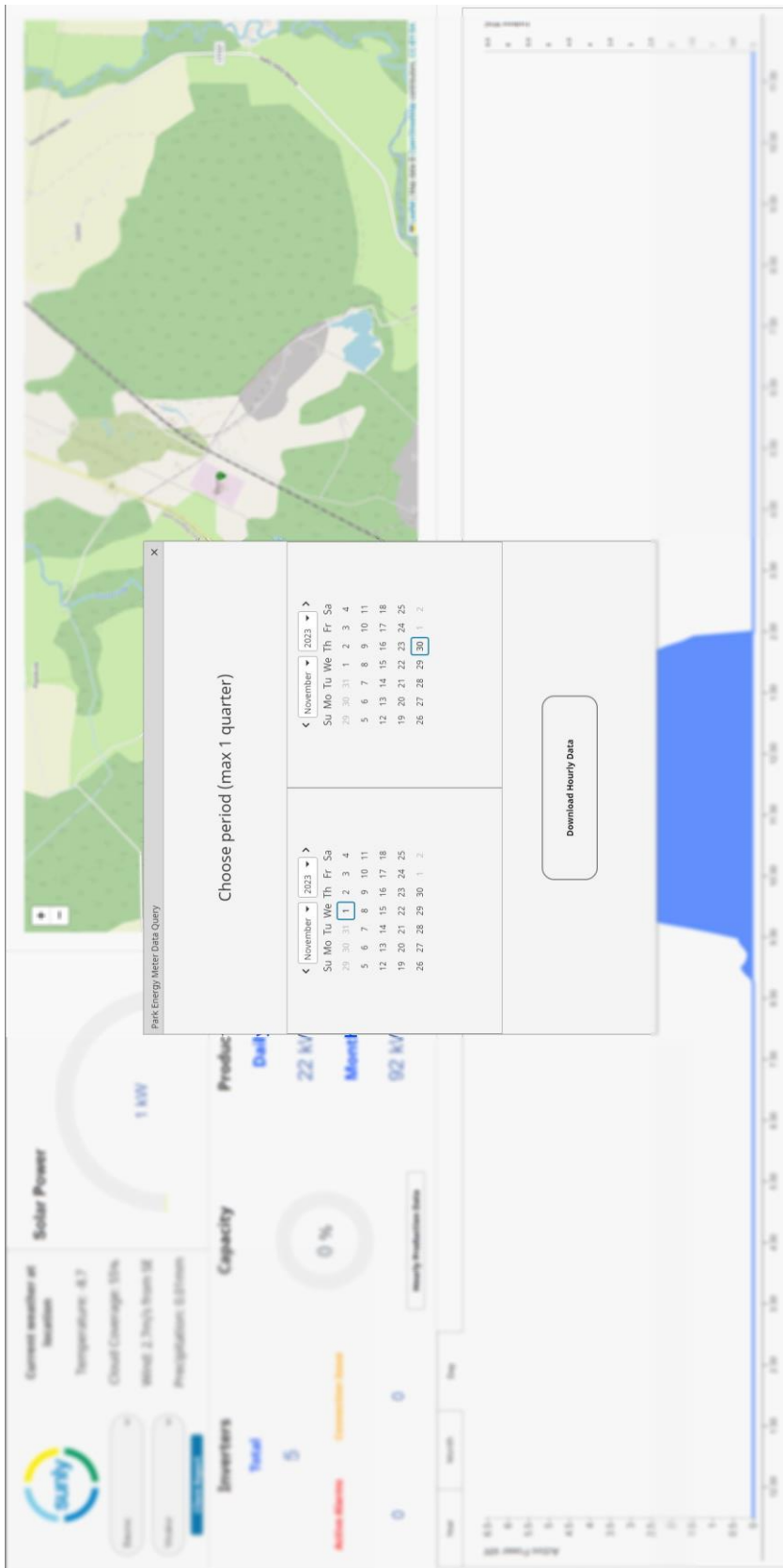


# Lisa 7. PowerChart tööriist





## Lisa 8. Hüpikaken tootmisandmete allalaadimiseks



## Lisa 9. Inverterite staatuse hüpikaken

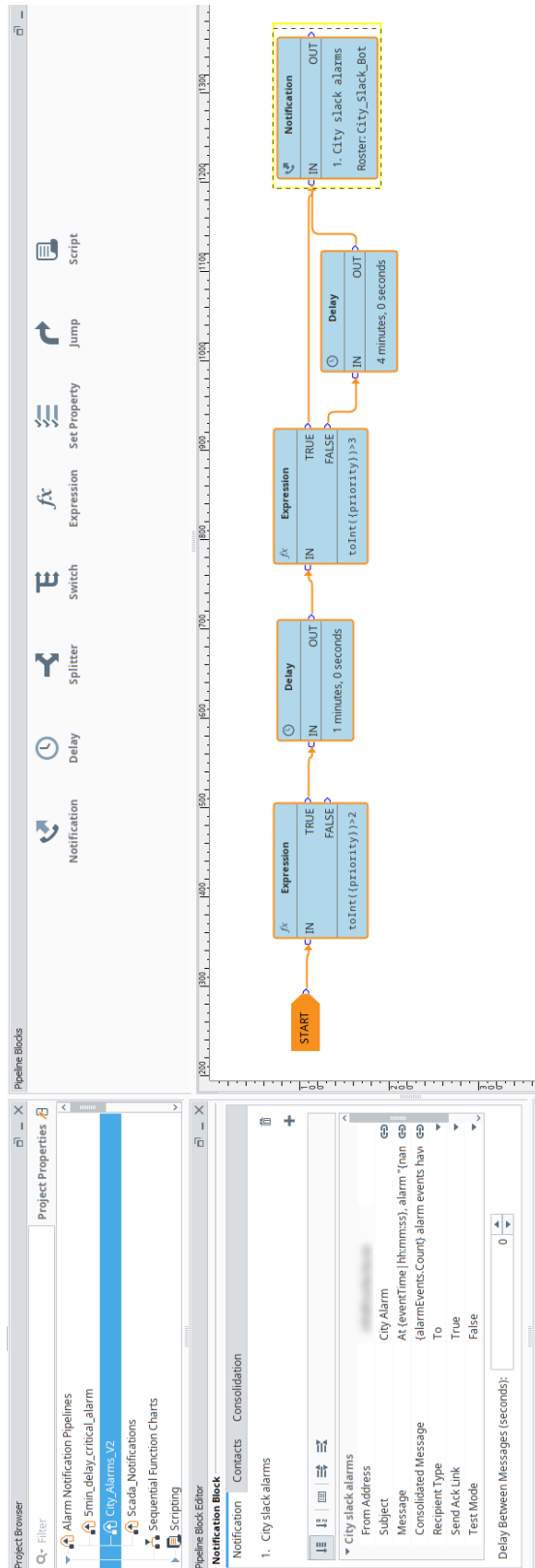
INVERTER STATUS REPORT

PROBLEM	NAME	TYPE
Communication down	[blurred]	Huawei
Communication down	[blurred]	Huawei
Communication down	[blurred]	Huawei

25 rows

1

# Lisa 10. Alarmist teavitamine kasutades Alarm Notification Pipeline moodulit



# Lisa 11. Kommunikatsiooniprobleemidest teadaandva alarmi seadistamine läbi tag-i

Tag Editor AF1\_Duration default

**Type Structure**

**Alarms**

Alarm Eval Enabled  True

Alarms

- Comms\_Down\_1h - Above Setpoint, High
- Comms\_Down\_4h - Above Setpoint, Critic

**Properties (Comms\_Down\_1h)**

Name	Comms_Down_1h
Enabled	true
Priority	High
Timestamp Source	System
Label	"Communication down for 1 hour at " + {PV_name} + "..."
Display Path	
Ack Mode	Manual
Notes	
Ack Notes Required	false
Shelving Allowed	true
<b>Alarm Mode Settings</b>	
Mode	Above Setpoint
Setpoint	3 600
Inclusive	true
Any Change	false
<b>Deadbands and Time Delays</b>	
Deadband	0
Deadband Mode	Absolute
Active delay (seconds)	0
Clear delay (seconds)	0
<b>Notification</b>	
Ack Pipeline	
Active Pipeline	Development/City_Alarms_V2
Clear Pipeline	

**Categories**

- All Properties
- Basic
- Meta Data
- 1a Value
- Numeric
- Security
- Scripting
- Alarms (2)
- History
- Custom

OK Apply Cancel

## Lisa 12. Kuuraporti raportipõhja esimene lehekülg



Hei,

Siin saad ülevaate oma päikesepargi kuuraportist

### Kokkuvõte Sinu päikesepargist



Päikesepaistelisi tunde



Elektritoodang



Toodangupõhine töökindlus



### Ilma kokkuvõte

Päikesepaiste

Sademed

aastate keskmisest

aastate keskmisest



### Inverterite ülevaade

Inverterite arv

Inverterite rikkeolukordade tõttu jäi tootmata

Keskmine muundusefektiivsus