

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Hugo Martin Teemus

**Reaalajas Delta keskuse õppe- ja teadushoone
andmete visualiseerimine**

Bakalaureusetöö (9 EAP)

Juhendaja: Pelle Jakovits, PhD

Tartu 2024

Reaalajas Delta keskuse õppe- ja teadushoone andmete visualiseerimine

Lühikokkuvõte:

Bakalaureusetöö eesmärk on uurida erinevaid lähenemisviise andmete visualiseerimiseks, analüüsida olemasolevat Delta keskuse õppe- ja teadushoone andmete visualiseerimise lahendust ning uurimise ja analüüsi tulemuste põhjal luua uus platvorm õppe- ja teadushoone andmete visualiseerimiseks. Töö esimene pool keskendub samas valdkonnas tehtud tööde ja olemasolevate lahenduste analüüsimisele ning uue platvormi nõuete kogumisele. Töö teises pooles käsitletakse valminud platvormi lahenduskäiku, analüüsitakse selle vastavust nõuetele ning tuuakse välja edasiarenduste võimalused.

Võtmesõnad:

Andmete visualiseerimine, veebirakendus, Cumulocity IoT, Inkscape, Next.js

CERCS:

P175 Informaatika, süsteemiteooria

Real-time visualization of Delta Centre's Study and Research Building data

Abstract:

The aim of the bachelor's thesis is to study approaches for data visualization, to analyze the existing solution for visualizing the Delta Centre's Study and Research building data and develop a new platform for visualizing the Study and Research building data based on the results of the research and analysis. The first half of the work focuses on analyzing the existing solutions relevant to the field of this thesis, and collecting the requirements for the new platform. The second part of the work includes the description of the development process and the analysis of the completed platform.

Keywords:

Data visualization, web application, Cumulocity IoT, Inkscape, Next.js

CERCS:

P175 Informatics, systems theory

Sisukord

Sissejuhatus	4
1. Taust	6
1.1 Andurid ja nutistu seadmed Delta keskuses	6
1.2 Cumulocity IoT	7
1.2.1 Cumulocity IoT API	8
1.3 InfluxDB	9
1.4 Grafana	10
1.5 Kaubanduskeskuste majaplaanid	10
1.6 Olemasolevad visualisatsioonid Delta keskusest	12
2. Olemasoleva lahenduse analüüs	15
2.1 Praeguse seisu analüüs	15
2.2 Nõuete kogumine	17
3. Lahenduskäik	18
3.1 Lahenduse arhitektuur	18
3.2 Korruste kuvandite loomine	20
3.2 Veebirakenduse funktsionaalsus ja interaktiivsus	26
3.2.1 Kuvandi liigutamine ja suurendamine	26
3.2.2 Andmete visualiseerimine kuvandil	27
3.2.3 Andmete ja kuvandite filtreerimine	29
3.2.4 Otsinguriba	29
3.2.5 Hüplikaken	30
3.2.6 Külgpaneel	30
3.2.7 Visualisatsioon ekraanile	31
3.3 Andmete pärimine ja edastamine	32
3.3.1 Temperatuuri ja CO2 andmed	33
3.3.2 Ajaplaanide andmed	34
3.3.3 Andmete edastamine	36
3.4 Platvormi ülesseadmine	36
4. Valminud platvormi analüüs	38
4.1 Vastavus nõuetele	38
4.3 Edasiarenduse võimalused	40
Kokkuvõte	42
Viidatud kirjandus	43
Lisad	46
Lisa 1. Platvormi lähtekood	46
Lisa 2. Litsents	47

Sissejuhatus

Tartu Ülikooli Delta keskus on 2020. aastal valminud hoonete kompleks, kuhu kuuluvad õppe- ja teadushoone (edaspidi Delta õppehoone) ning ettevõtlusmaja. Hoonetes on paigaldatud erinevaid seadmeid ja andureid, mis koguvad andmeid. Lisaks on seadmetel ja anduritel interneti ühenduvuse võimekus, mille kaudu on võimalik seadmeid konfigureerida ja reaalajas edastada andmeid interneti teel. 2024. aasta seisuga hallatakse seadmeid Cumulocity IoT platvormi kaudu, mis võimaldab määrata seadmetele nimetusi ja korraldada hierarhilist struktuuri, kus on täpsustatud ülem- ja alamseadmete vahelised suhted. Hetkel kasutatakse reaalajas kogutud andmete visualiseerimiseks Grafana platvormi, mis võimaldab luua mitmekülgseid näidikupaneele.

Praeguses lahenduses esineb puudujääke. Grafana platvormis on õppehoone ruumide ning ruumis olevate seadmete suhted iga ruumi kohta käsitsi konfigureeritud ning puudub tsentraalne konfiguratsioon, mis sisaldaks ruumide ja seadmete vahelisi suhteid. Seadistuste muutmiseks peab käima iga ruumi numbri kohta loodud päringud eraldi üle, mistõttu ei ole võimalik seadistusi visualiseerida ega teha muudatusi kiiremalt ja mugavamalt tsentraalsest kohast. Andmeid visualiseeritakse Grafana platvormis koos vastava ruumi numbriga, kuid puuduvad visualiseeringud asukoha põhjal, näiteks andmete kuvamine majaplaanil vastavalt andmeid kogunud seadme asukoha põhjal. Praegune lahendus ei ole kasutajasõbralik olukorras, kus kasutaja soovib andmete visualiseeringut asukoha põhjal, kuid ei tea soovitud asukohas olevate ruumide nimetusi.

Bakalaureusetöö eesmärk on uurida võimalikke lähenemisi andmete visualiseerimiseks ning luua Delta õppehoone andmete visualiseerimise platvorm. Töö käigus uuritakse samas valdkonnas tehtud töid ja olemasolevaid lahendusi, et analüüsida erinevaid lähenemisi. Lisaks analüüsitakse õppehoone olemasolevat lahendust, et kaardistada lahenduse head küljed ja probleemkohad. Analüüside ja kogutud nõuete põhjal tehakse loodava platvormi tehnoloogilised ja disaini valikud. Eesmärk on luua kergesti ligipääsetav platvorm, kus visualiseeritakse Delta õppehoone andmeid kasutajale võimalikult arusaadavalt ja kasutajasõbralikult.

Bakalaureusetöö esimeses peatükis antakse ülevaade töö käigus käsitletud teemadest ja tehnoloogiatest ning valdkonnaga seotud töödest ja lahendustest. Teises peatükis analüüsitakse Delta õppehoone olemasolevat lahendust ning kirjeldatakse uue platvormi jaoks kogutud nõudeid. Kolmandas peatükis kirjeldatakse platvormi arhitektuuri ja

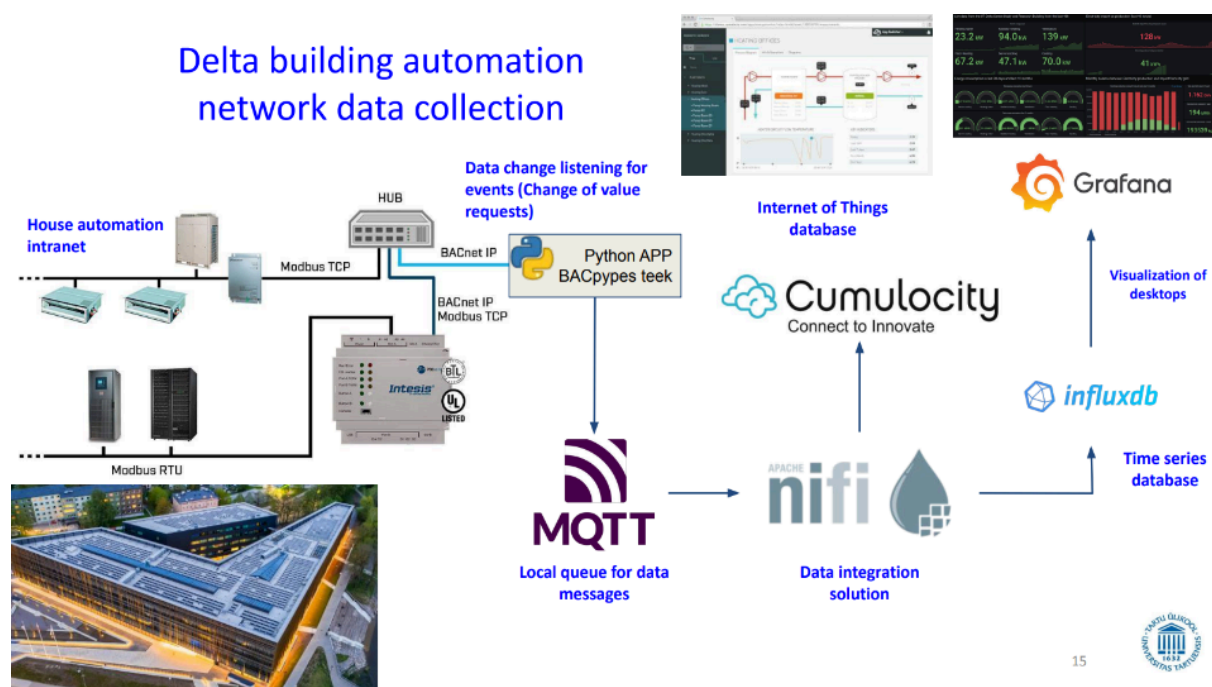
arendusprotsessi. Neljandas peatükis analüüsitakse platvormi vastavust nõuetele ning kirjeldatakse võimalikke edasiarendusi.

1. Taust

Järgnevas peatükis tutvustatakse teemasid ja tehnoloogiad, millest peaks olema lugejal ülevaade, et käesolevast tööst aru saada. Lisaks keskendutakse olemasolevatele lahendustele ja uurimistöödele, mis on selles valdkonnas varem tehtud. Eesmärk on anda lugejale põhjalik ülevaade hoone andmete visualiseerimise teemal tehtud töödest ja lahendustest ning analüüsida nende tööde ja lahenduste tehtud valikuid disaini- ja funktsionaalsuse osas.

1.1 Andurid ja nutistu seadmed Delta keskuses

Delta keskus on Tartu Ülikoolile kuuluv hoonete kompleks, kuhu kuuluvad õppehoone ning ettevõtlusmaja [1]. Hoonetesse on paigaldatud andureid ja seadmeid, mis mõõdavad mitmesuguseid näitajaid. Eelmainitud seadmed kuuluvad Delta keskuse nutistusse ehk asjade internetti [2], mis tähendab, et seadmed omavad funktsionaalsust lokaalse võrgu kaudu omavahel suhelda ja edastada andmeid (vt joonis 1) [3].



Joonis 1. Andmete kogumine Delta keskuses [4].

Mõned näited Delta keskusesse paigaldatud nutistu seadmetest on a) ventilatsiooni- ja kütteseadmed, mis mõõdavad energiakulu, b) klassiruumidesse paigaldatud CO2

(süsinikdioksiidi) kontsentratsiooni ja temperatuuri mõõtvad andurid ja nutikad termostaadid, mis edastavad andmeid seadistatud kütterežiimist (vt joonis 2)¹.

AK01'Vent'SV2'Vav3	AK01'Vent'SV2'Vav3'RuumTESP2.X6	('analogValue', 148) CO2 tase 6
	AK01'Vent'SV2'Vav3'RuumTESP2.NumVldXY	(202, 6) Num.valid sample pnts X,Y
	AK01'Vent'SV2'Vav3'TEpeRoot	('trendLog', 38) Trend
	AK01'Vent'SV2'Vav3'TSu2	('analogInput', 30) QE1020 Ruumitemperatuuri andur
	AK01'Vent'SV2'Vav3'DP	('analogInput', 27) QE1020 CO2 andur
	AK01'Vent'SV2'Vav3'HuSu	('trendLog', 36) Trend
	AK01'Vent'SV2'Vav3'RuumTESP2.Out	('analogValue', 158) Output
	AK01'Vent'SV2'Vav3'Mdlt12	('analogInput', 29) VAV VT 1020.2 tagasiside
	AK01'Vent'SV2'Vav3'TEpeRoot1	('trendLog', 39) Trend

Joonis 2. Delta keskuse andurite ja seadmete nimetuste tabel.

Delta keskuse nutistu kaudu edastatakse reaajas mõõtmelid hoone automaatika, Cumulocity IoT ja InfluxDB andmebaasidesse. Andmebaasi abil saab kogu andmestikku hallata ühest kohast. Andmeid kasutatakse paigaldatud seadmete seisundite visualiseerimiseks ning sisendina erinevates TÜ-s läbiviidavates projektides².

1.2 Cumulocity IoT

Cumulocity IoT (versioon 10.18.0) (edaspidi Cumulocity) platvorm võimaldab kasutajal ühendada nutistu seadmed pilvekeskkonnaga, kus saab kõiki ühendatud seadmeid hallata ja nende seisundeid visualiseerida ühes kohas [5]. Platvormis saab määrata seadmete hierarhiat ning grupeerida seadmeid, mis muudab seadmete haldamist lihtsamaks (vt joonis 3). Lisaks saab mõõdetud andmeid platvormis visualiseerida graafikutel (vt joonis 4).

AK01'Vent'SV2 Showing 18 child devices

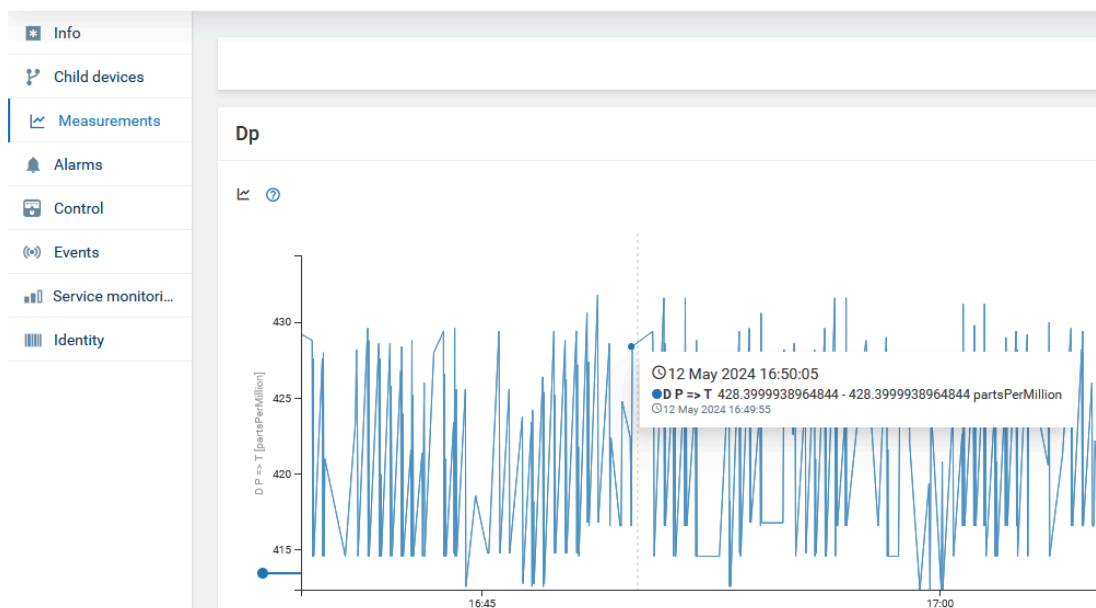
Info	STATUS	NAME	MODEL	SERIAL NUMBER	REGISTRATION DATE	SYSTEM ID
Child devices	≡	AK01'Vent'SV2'Ccl			26 Jul 2022, 15:09:28	151
Measurements	≡	AK01'Vent'SV2'DmpShofOa			26 Jul 2022, 15:09:28	153
Alarms	≡	AK01'Vent'SV2'Erc			26 Jul 2022, 15:09:28	154
Control	≡	AK01'Vent'SV2'FanEx			26 Jul 2022, 15:09:28	157
Events						

Joonis 3. Näide Delta õppehoone seadmete hierarhiast Cumulocity keskkonnas³.

¹ Näited pärinevad töö autorile juhendaja poolt 5. oktoobril 2023. aastal e-kirja teel saadetud tabelist, kus on ülevaade Delta keskuse hoonetesse paigaldatud anduritest ja seadmetest.

² Andmete kasutusvaldkonnad selgusid töö autori ja juhendaja suulise arutelu käigus.

³ Näide pärineb Delta keskuse Cumulocity keskkonnast.



Joonis 4. Mõõdetud andmete visualiseerimine graafikul Cumulocity platvormis⁴.

Delta keskuse nutistu seadmete haldamiseks kasutatakse Cumulocity platvormi.

1.2.1 Cumulocity IoT API

Cumulocity funktsionaalsused on kättesaadavad rakendusliidese kaudu. Rakendusliides (ingl *Application Programming Interface*) (edaspidi API) on reeglite, funktsioonide ja protokollide kogum, mille abil saavad tarkvararakendused omavahel suhelda [6]. Cumulocity API toetab internetiaadressi kaudu andmete pärimist ning teavituste edastamist [7]. Cumulocity keskkonnas on igale seadmele omistatud “id” väärtused, mille põhjal saab API kaudu seadme andmeid pärida. Näiteks reaalaaja andmete kättesaamiseks saab “id” väärtuse põhjal pärida seadme poolt kindla ajavahemiku sees mõõdetud andmeid.

Kirjeldatud päringu internetiaadress:

<https://delta.iot.cs.ut.ee/measurement/measurements?source=176&dateFrom=2024-04-30T15:42:00.254Z&dateTo=2024-04-30T15:43:00.254Z&valueFragmentType=TSu2>

⁴ Graafik pärineb Delta keskuse Cumulocity keskkonnast.

Parameeter	Väärtuse kirjeldus
source	seadme "id" väärtus Cumulocity keskkonnas
dateFrom	ajavahemiku algus, ISO 8601 formaadis
dateTo	ajavahemiku lõpp, ISO 8601 formaadis
valueFragmentType	seadme andmetüübi

Tabel 1. Päringu parameetrite kirjeldused.

```
"measurements": [
  {
    "self": "https://t10162.iot.cs.ut.ee/measurement/measurements/557011229",
    "time": "2024-04-30T15:42:01.000Z",
    "id": "557011229",
    "source": {
      "self": "https://t10162.iot.cs.ut.ee/inventory/managedObjects/176",
      "id": "176"
    },
    "type": "DP",
    "DP": {
      "T": {
        "unit": "partsPerMillion",
        "value": 471.800018310547
      }
    }
  }
],
```

Joonis 5. Päringu vastus.

Tabel 1 kirjeldab näidispäringu parameetrite väärtusi ning joonis 5 kujutab näidispäringu vastuse struktuuri ja sisu.

1.3 InfluxDB

InfluxDB (versioon 2.7.1) on InfluxData poolt loodud avatud lähtekoodiga andmebaasi platvorm, mis on üles ehitatud talletama aegridade (ingl *time series*) andmestikku [8]. Aegread on andmed, mis on indekseeritud ajalisel järjekorras [9]. Delta keskuse näitel on aegread kindlatel ajahetkedel mõõdetud ruumi temperatuuri väärtused. Võrreldes Cumulocity platvormi andmebaasiga, toetab InfluxDB andmebaas mitmekülgsemate ajalisi piiranguid käsitlevate päringute loomist ning andmebaasist andmete pärimine koormab InfluxDB andmebaasi vähem⁵. Eelmainitud omaduste tõttu edastatakse Delta keskuses mõõdetud andmed lisaks Cumulocity andmebaasile ka InfluxDB andmebaasi.

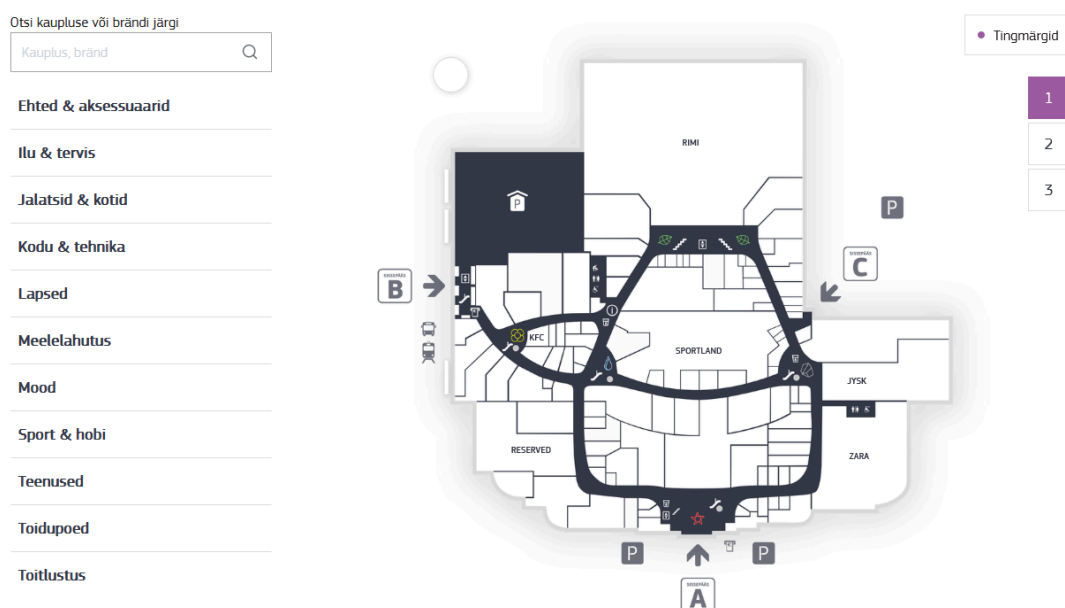
⁵ Kirjeldatud eelised selgusid töö autori ja juhendaja suulise arutelu käigus.

1.4 Grafana

Grafana (versioon 10.1.5) on avatud lähtekoodiga platvorm, mis võimaldab päringute põhjal luua mitmekülgseid visualisatsioone andmestikust [10]. Andmeid saab visualiseerida eri tüüpi graafikutel ja diagrammidel [11]. Keskkonnas saab luua näidikupaneele (ingl *dashboard*), kus saab kasutaja enda nõuete järgi lisada ja kujundada erinevaid visualisatsioone ühele veebilehele [12]. Lisaks on Grafanal funktsionaalsus saata välja hoiatusi kasutaja poolt loodud reeglite järgi ning jooksutada päringuid andmestiku uurimiseks [13, 14]. Unwin [15] leiab, et andmete visualiseerimine graafikutel aitab ära tunda andmetes trende ning leida anomaaliad. Delta keskuse andmestiku visualiseerimiseks kasutatakse Grafana abil loodud graafikuid.

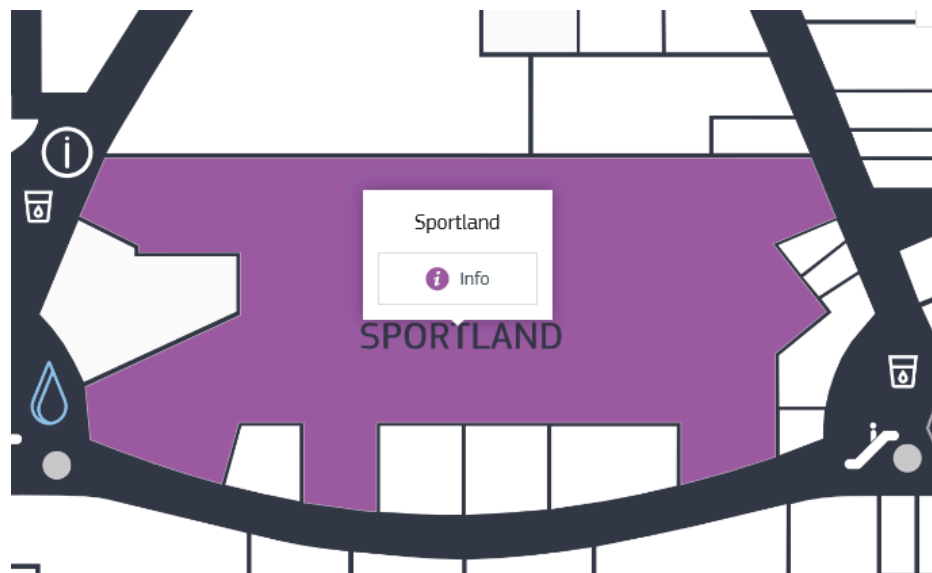
1.5 Kaubanduskeskuste majaplaanid

Mitmetel kaubanduskeskustel on olemas interaktiivsed majaplaanid, kus on kujutatud hoone korruseplaanid kas kahe- või kolmedimensioonilises (edaspidi 2- ja 3D) vaates. Majaplaane kuvatakse näiteks kaubanduskeskustes asuvatel infotabloodel ja keskuste veebilehtedel. Majaplaani eesmärk on lihtsustada kaubanduskeskuse kliendile keskuses asuvate kaupluste ja pakutavate teenuste kohta informatsiooni leidmist. Kaupluste ja teenuste kohta kuvatakse näiteks nende asukohti kaubanduskeskuses, nende tegevusvaldkondi, lahtiolekuaegu ja kontaktandmed. Näiteks on kaubanduskeskuse Ülemiste veebilehel [16] olemas interaktiivne majaplaan, mida on kujutatud 6. joonisel.



Joonis 6. Kaubanduskeskuse Ülemiste interaktiivne majaplaan [16].

Majaplaanile on ligipääsemiseks vaja ainult internetiühendust ja veebibrauserit toetavat seadet, tänu millele on majaplaan kaupluse klientidele laialdaselt kättesaadav. Majaplaan on kujutatud 2D-s pealtvaates. Majaplaan on interaktiivne, mida on võimalik suurendada ja ringi liigutada. 2D võib olla kliendile 3D vaatest intuitiivsem, kuna majaplaani ringi liigutamiseks on vaja vähem juhtnuppe ning puutežeste. Plaani suurendamise ja liigutamise funktsionaalsus muudab majaplaani navigeerimise ligipääsetavamaks näiteks kasutajale, kellel on raskusi plaanilt detailide välja lugemisega algse suurendusega formaadis. Ruume on võimalik valida majaplaanil ruumi peale vajutades, kaupluse nimetuse järgi otsinguriba kasutades või otsinguriba all oleva kaupluste nimekirja põhjal. Ruumi valimine muudab valitud ruumi värvuse lillaks ning ruumi keskele tekitatakse väike hüpikaken, kus kuvatakse ruumis asuva kaupluse nimetus ja nupp, mida vajutades avaneb kaupluse kohta lisainformatsiooni kuvav brauseri vaheleht (vt joonis 7).

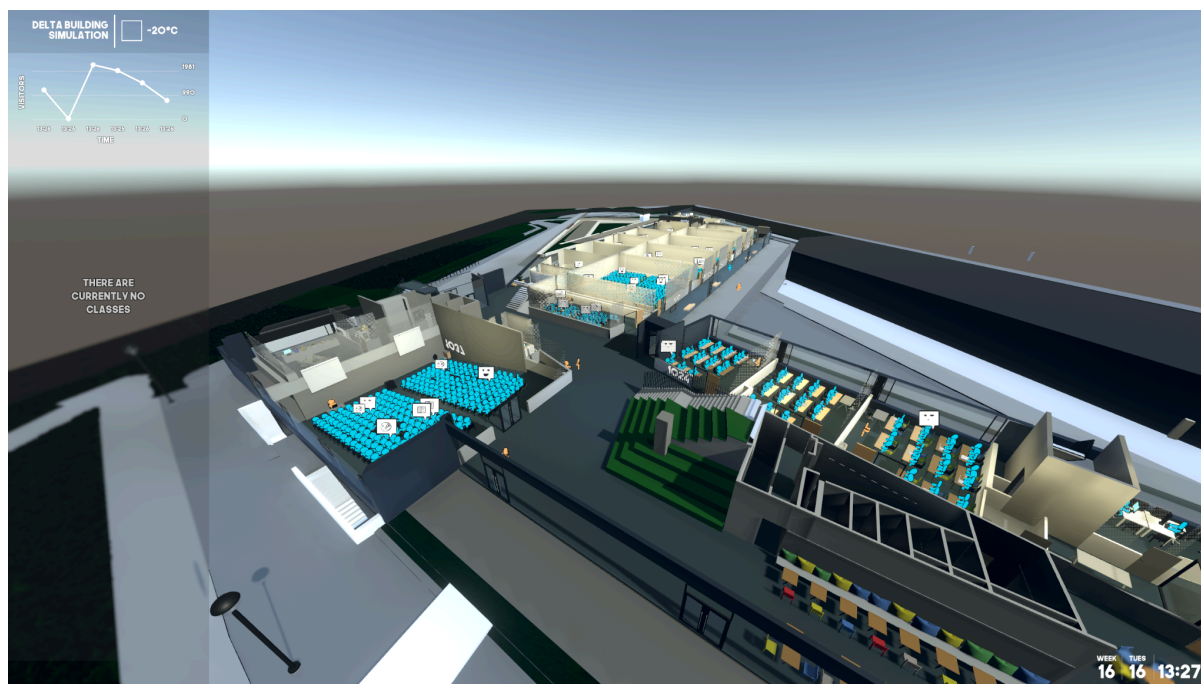


Joonis 7. Valitud ruumi keskele tekkiv hüpikaken [16].

Kirjeldatud viisil kaupluse asukoha otsimine on kasutajasõbralik. Antud näites on valitud ruumis asuva kaupluse nimetusel selge visuaalne ühendus valitud ruumi asukohaga majaplaanil, tänu millele on kasutajale kaupluse asukoha otsimine intuitiivne. Keskuses asuvate üldteenuste ja rajatiste asukohad on majaplaanile märgitud tingmärkidena. Näitena valiti kaubanduskeskuse Ülemiste veebilehel asuv lahendus, kuna see on klientidele kättesaadav ning tänu lahenduses tehtud disaini- ja funktsionaalsuse valikutele on majaplaani navigeerimine intuitiivne.

1.6 Olemasolevad visualisatsioonid Delta keskusest

Delta õppehoonest on loodud mitmeid visualiseerimisprojekte. Alates 2017. aastast on TÜs Arvutigraafika ja virtuaalreaalsuse labori juhendamisel tudengite poolt arendatud 3D mudeli baasil Delta õppehoone visualiseerimist [17]. Esimene kirjeldatud visualiseerimist käsitleva platvormi nimi on Delta Building Visualization (edaspidi DBV). DBV on Unity mängumootoriga loodud õppehoone visualiseerimisplatvorm, milles kasutatakse arhitektuurfirma Arhitekt11 OÜ poolt loodud õppehoone 3D mudelit [18]. Joonisel 8 on kujutatud DBV platvormis olevat õppehoone mudelit ja platvormi kasutajaliidest.



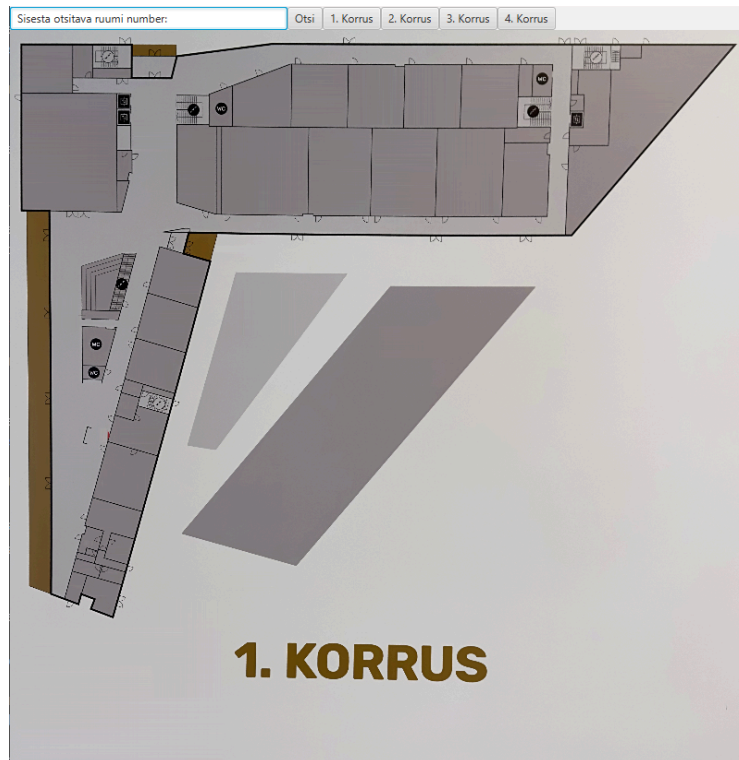
Joonis 8. Kuvatõmmis Delta Building Visualization platvormist [17].

Lisaks õppehoone 3D mudelile, DBV simuleerib reaalajas õppehoone ruumide täituvusi ruumi ajaplaanide põhjal ja hoone ümber olevat ilmastikku [17]. DBV arendamises on osalenud üheksa TÜ tudengit ning projekti erinevate funktsionaalsuste edasiarendustest on valminud kuus bakalaureusetööd. DBV projekti baasil arendatakse 2020. aastast alates DeltaVR platvormi [19]. DeltaVR on Delta õppehoone visualiseerimisplatvorm, mis toetab virtuaalreaalsus (edaspidi VR) seadmete kasutamist visualisatsioonis navigeerimiseks. 3D mudeli kasutamisest tuleneb Delta õppehoone visualisatsiooni platvormidel puudujääke, mida sooviti loodud platvormiga lahendada.

- DBV ja DeltaVR platvormide kasutamiseks on esmalt vaja alla laadida platvormi failid ning seejärel need failid lahti pakkida. Olenevalt interneti kiirusest ja süsteemi mäluseadme jõudlusest, võib kokkupakitud failide allalaadimine ja nende lahti pakkimine võtta mitmeid minuteid aega. Töö käigus valminud platvorm on loodud ligipääsetavuse kiirust ja ulatuslikkust silmas pidades. Platvorm asub veebikeskkonnas, seega platvormi veebirakenduse kasutamiseks on vaja internetiühendust ja veebibrauserit toetavat seadet, et veebirakendusele ligi pääseda mõne sekundiga.
- DBV ja DeltaVR kasutavad Unity mängumootorit. Platvormid vajavad stabiilse ja kasutajasõbraliku kaadrisageduse saavutamiseks piisavalt riistvaralist jõudlust. Liiga madala kaadrisageduse korral ei ole platvorm kasutajasõbralik. Madalal kaadrisagedusel ei pruugi rakendus olla suuteline registreerima järjepidevalt kasutaja poolt antud sisendeid. Lisaks on kaadrite vahetumine madalal kaadrisagedusel liiga katkendlik, et kasutajal saaks olla sujuv kogemus.

Arvestades peatükis 2.2 uue lahenduse kogutud nõudeid, töö autori vähest kogemust 3D programmidega ja eelpool kirjeldatud 3D mudeli baasil loodud platvormide arendamiseks kirjeldatud ajakulu ja arendajate hulka [17], otsustati platvormi loomisel kasutada 2D visualisatsioone.

Delta õppehoonest on ka 2D pealtvaates korruste plaanide baasil loodud hoone visualiseerimist käsitlev programm nimega Delta Majaplaan. Programm on loodud kahe TÜ tudengi poolt aine “Objektorienteeritud Programmeerimine” raames [20]. Joonisel 9 on kujutatud Delta Majaplaan platvormi kasutajaliidest.



Joonis 9. Delta Majaplaan platvormi kasutajaliides [20].

Programm on loodud JavaFX-iga ning on eraldiseisev, mis tähendab, et programmi kasutamiseks on vaja koodi repositooriumist programmi failid alla laadida. Selle tõttu on võimalik programmi kasutada ainult arvutiga. Kogu funktsionaalsus ja informatsioon on talletatud programmi failidesse ning programmi kasutamiseks ei ole vajalik internetiühenduse olemasolu. Programmi kasutajaliides koosneb otsinguribast, nuppudest korruste valimiseks ja valitud korruse pealtvaate kuvandist [20]. Ruume saab leida ja valida kasutades otsinguriba, kuhu ruumi numbri sisestamisel kuvatakse ruumi asukoht hoones. Lisaks saab ruume valida ruumi peale vajutades. Valitud ruumi kohta kuvatakse informatsiooni. Programm salvestab kasutaja tegevused ja sisendeid logifaili, mille abil on programmi arendajate sõnul võimalik analüüsida näiteks enim otsitud ruume hoones [20].

Võrreldes Delta Majaplaan programmiga, on bakalaureusetöö käigus arendatud platvorm ligipääsetavam. Platvormile pääseb ligi veebibrauseriga ning ei vaja toimimiseks lisafailide allalaadimist. Erinevalt Delta Majaplaan programmiga, on vaja platvormi kasutamiseks internetiühendust. Internetiühendus võimaldab platvormis visualiseerida lisaks hoone korruste plaanidele ka dünaamiliselt ruumide reaalaaja andmeid.

2. Olemasoleva lahenduse analüüs

Järgnevas peatükis analüüsitakse olemasolevat lahendust, et kaardistada lahenduse head küljed ja probleemkohad. Lisaks tutvustatakse uue lahenduse nõudeid, mille eesmärk on lahendada olemasoleva lahenduse puuduseid.

2.1 Praeguse seisuga analüüs

Delta õppehoones on 2024. aasta seisuga kasutusel nutistu seadmete poolt kogutud andmete visualiseerimiseks Grafana keskkonnas loodud graafikud. Delta õppehoone Grafana keskkonnale on ligipääs ainult Grafana keskkonna edasiarendusega tegelevatel isikutel. Avaliku ligipääsu puudumine keskkonnale tagab andmete turvalisuse ja keskkonna töökindluse, kuna andmetele on juurdepääs piiratud, ning võimalus, et keegi halvab keskkonna tööd, on madalam. Suuremale inimgrupile, näiteks TÜ õpilastele ligipääsu tagamine Grafana visualisatsioonidele vajaks tarkvaralise lahenduse juurde loomist. Võimalik lähenemine arendusele oleks veebilehe loomine, mis on kas avalikult või ainult TÜ kasutajaga sisse logides ligipääsetav ja kus kuvatakse Grafana keskkonnas loodud näidikupaneele.

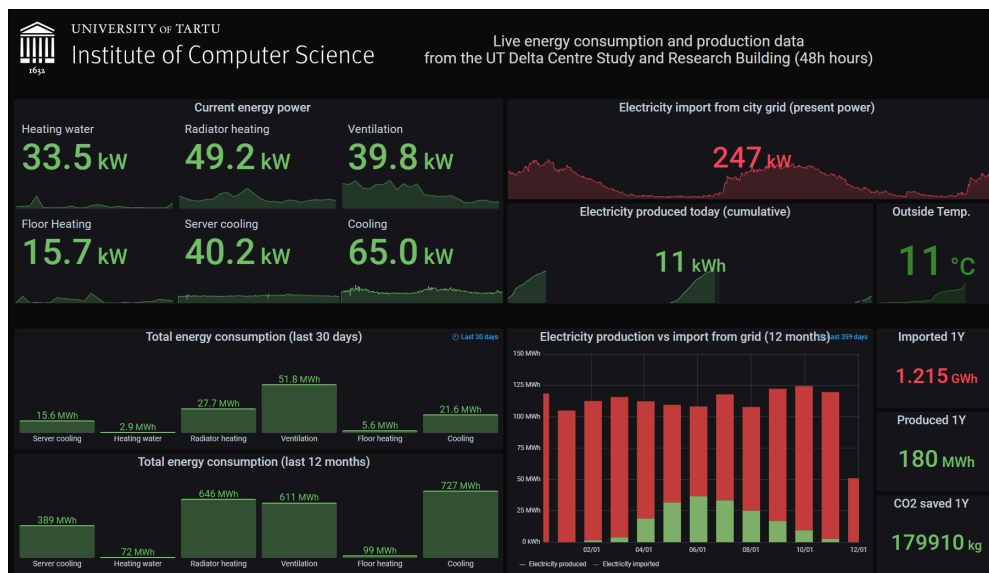
Olemasolevatel Grafana visualisatsioonidel kujutatud väärtused ei pruugi olla lihtsalt arusaadavad. Näiteks CO₂ kontsentratsiooni väärtused, millel puuduvad praeguses lahenduses selgitused, peaks visualiseerima koos selgitusega. Vastasel juhul võib kasutajale arusaamatuks jääda, kas väärtused on kõrged või madalad. Lin jt [21] on leidnud, et andmete visualiseerimisel mängivad olulist rolli värvid. Lin toob välja, et värvide abil saab inimestele väljendada andmete väärtuste tähendusi assotsiatsioonide kaudu. Grafana keskkonnas on värvide kasutamise funktsionaalsus olemas [22] ning olemasolevas lahenduses kasutatakse seda funktsionaalsust.

Praegusel lahendusel puudub integratsioon ülejäänud TÜ pilvekeskkondadega. Näiteks on TÜ-l olemas õppeinfosüsteem ÕIS 2⁶, kus on olemas ruumide ajaplaanid, ning Delta õppehoone õpilastele mõeldud koostööruumide jaoks on loodud broneerimissüsteemi platvorm DeltaQR. Ajaplaanide sidumine andurite andmestikuga aitaks paremini visualiseerida ruumide täituvust. Sánchez jt [23] leidsid analüüsi käigus, et CO₂ kontsentratsiooni põhjal on võimalik järeldada klassiruumi inimtäituvust. Seega on võimalik kasutada koos ruumi CO₂ kontsentratsiooni andmeid ja ruumi ajaplaani andmeid, et arusaadavalt visualiseerida, kas valitud ruum on parasjagu hõivatud või mitte.

⁶ <https://ois2.ut.ee>

Praeguses lahenduses saab mõõdetud andmeid otsida anduri nime või ruumi numbri järgi. Ainult teksti järgi andmete otsimine ei ole kasutajasõbralik lahendus. Praegune lahendus ei võimalda kasutajal otsida andmeid olukorras, kus kasutaja teab ainult otsitava ruumi asukohta hoones, kuid ei tea ruumi numbrit. Näiteks kasutavad Román-García jt [24] oma lahenduses andmete visualiseerimiseks lisaks Grafana keskkonnas loodud graafikutele ka hoone 3D mudelit. Tema lahenduses on mudelilt võimalik näha andmeid ruumi ja andmetüübi järgi. Eelnevalt kirjeldatud lähenemine on kasutajasõbralikum Delta keskuse andmestiku visualiseerimisest, kuna andmeid saab otsida nii ruumi numbri kui ka ruumi asukoha järgi.

Olemasoleva lahenduse puhul on Delta õppehoones kättesaadav Grafana keskkonnas loodud näidikupaneel, mida kuvatakse perioodiliselt õppehoone fuajees asuvatel ekraanidel. Näidikupaneelil visualiseeritakse Delta õppehoone energiakulu ja -tootlikkust (vt joonis 10).



Joonis 10. Delta õppehoone energiakulu ja -tootlikkust visualiseeriv Grafana näidikupaneel⁷.

Näidikupaneelil visualiseeritakse andmete reaalaaja väärtusi koos graafikutega, mis kujutavad viimase 48 tunni väärtusi. Kirjeldatud visualiseerimisest saab kasutaja lisaks reaalaaja väärtusele ka ülevaate üldisemast trendist, mis võib omada väärtuslikku lisainformatsiooni. Näiteks saab kasutaja kõrge reaalaaja väärtuse korral graafiku põhjal kiirelt järeldada, kas väärtus vastab trendile või on tegemist anomaaliaga.

⁷ Ekraanivaade näidikupaneelist pärineb töö autorile juhendaja poolt 1. detsembril 2023. aastal saadetud e-kirjast.

2.2 Nõuete kogumine

Uue lahenduse nõuded tulenevad töö autori suhtlusest juhendajaga ja olemasoleva lahenduse analüüsist.

- Olemasolevas lahenduses saab ruumis kogutud andmeid otsida ainult ruumi numbri järgi. Uues lahenduses peab saama ruumis kogutud andmeid otsida ruumi numbri ja asukoha põhjal. See tähendab, et lahenduses oleks lisaks ruumi numbri järgi andmete pärimise funktsionaalsusele olemas ka visualisatsioon Delta õppehoone ruumide paigutusest. Visualisatsioon tagab, et andmeid saab näha ka andmete asukoha järgi olukorras, kus kasutajal puudub informatsioon ruumide asukohtadest. Lisaks täidab õppehoone visualisatsioon ka navigeerimistööriista rolli, mis võimaldab otsida ruumi asukohta õppehoones ruumi numbri järgi.
- Uus lahendus peab olema avalikult ligipääsetav ilma lisafailide allalaadimiste ja rakenduste installeerimisega. Platvorm luuakse veebikeskkonda, millega tagatakse, et platvormile ligi pääsemiseks on vaja ainult internetiühendust ja veebibrauseri programmi. Lisaks tagab veebikeskkonna baasil loodud platvorm ühilduvuse erinevate seadmetega nagu arvutid, nutitelefonid ja tahvelarvutid.
- Uues lahenduses peab olema tagatud kasutajasõbralik kogemus ka nõrgema jõudlusega seadmetel. TÜ arvutiteaduse instituudist (edaspidi ATI) laenutatavad sülearvutid peaksid võimaldama platvormi kasutamisel kasutajasõbraliku kogemuse, mis tagab, et ATI tudengid saavad lihtsalt ja mugavalt platvormi kasutada.
- Uues lahenduses peab olema funktsionaalsus tekitada visualisatsioon, milles on visualiseeritavat korrust ja andmetüüpi võimalik filtreerida internetiaadressisse sisestatud filtreeringute järgi. Sisestades filtreeringud internetiaadressi lõppu, ei pea visualisatsiooni pärast veebilehe laadimist soovitud filtreeringute jaoks ümber seadistama. Funktsionaalsus võimaldab lihtsalt üles seada visualisatsioone ilma sisendseadmeteta nagu arvutihiire ja -klaviatuuri või puutetundliku ekraaniga. Näiteks kuvatakse Delta õppehoone esimese korruse fuajees asuvatel ekraanidel perioodiliselt internetiaadressi järgi Grafana keskkonnas loodud näidikupaneeli.
- Uus lahendus peab olema jätkusuutlik. See tähendab, et platvormi lähtekood on koodi repositooriumist avalikult kättesaadav, koodi on lisatud selgitavad kommentaarid ja koodi repositoorium sisaldaks platvormi töövooskeemi. Loetletud tingimused tagavad pärast bakalaureusetöö valmimist teiste isikute suutlikkuse platvormi edasi arendada.

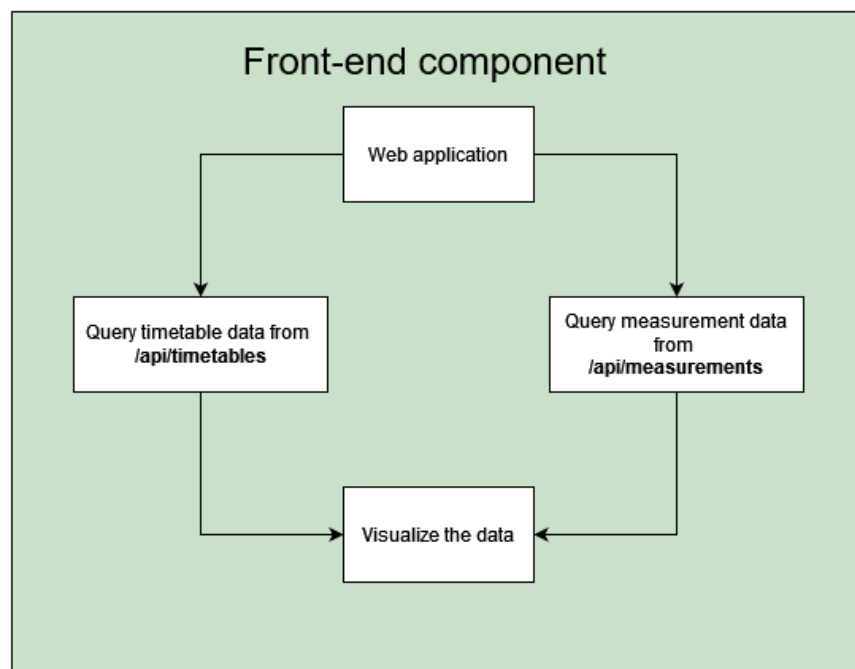
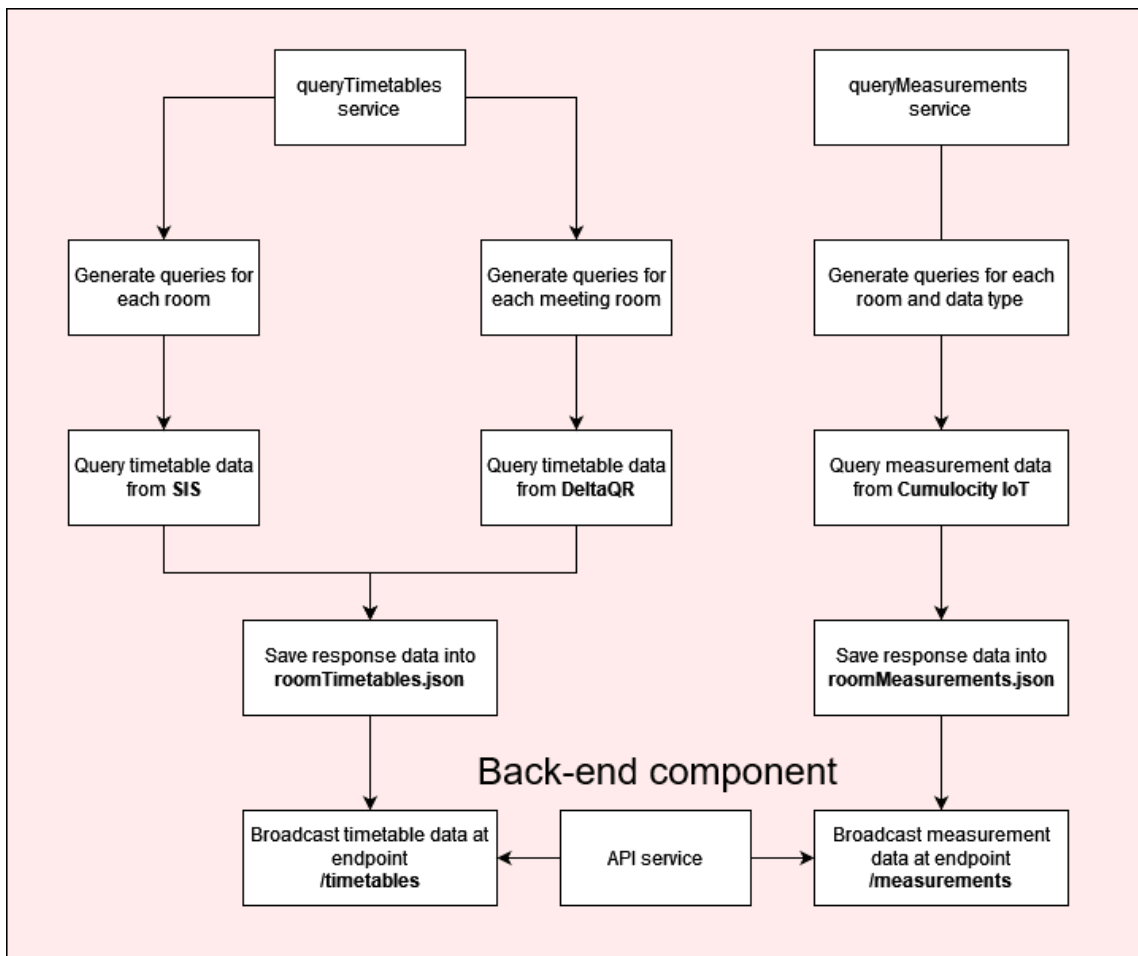
3. Lahenduskäik

3.1 Lahenduse arhitektuur

Platvormi arendamise käigus otsustati, et platvorm jaguneb kaheks komponendiks.

- Esimeses komponendis asub veebirakendus, mis käsitleb andmete visualiseerimist.
- Teises komponendis asuvad andmete pärimise teenused ja API teenus andmete edastamiseks veebirakendusele.

Iga seadme ja ruumi kohta on vaja teha eraldi päringud. Andmete pärimine ja töötlemine on mõistlikum teha serveripoolselt, et veebirakenduse koodis peaks tegema võimalikult vähe päringuid, et saavutada veebirakenduse parem jõudlus. Lisaks muudab platvormi koodi organiseerimine komponentideks koodi loetavust ning lihtsustab vigade tuvastamist. Joonis 11 kujutab platvormi vooskeemi.



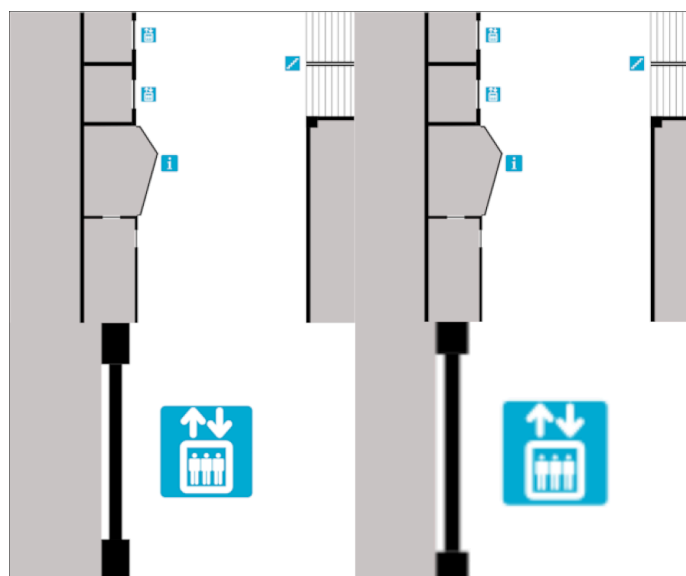
Joonis 11. Platvormi vooskeem.

3.2 Korruste kuvandite loomine

Delta õppehoones asuvad kolmandal ja neljandal korrusel kontoriruumid. Platvormi arendamise ajal koguti üksikute kolmanda ja neljanda korruse ruumide kohta andmeid. Kontoriruumide andmete reaajas visualiseerimine avalikult kättesaadavas platvormis võib rikkuda inimeste privaatsust. Delta õppehoones on kontoriruumid määratud kahele või enamale inimesele töötamiseks. Näiteks saab kontoriruumi CO₂ kontsentratsiooni põhjal järeldada, kas inimene on parasjagu kontoris. Arvestades väheste andmetega ja isikute privaatsusega, otsustati, et platvormis visualiseeritakse ainult esimese ja teise korruse ruumide andmeid.

Kõigepealt oli vaja otsustada, millisel viisil hakatakse platvormis Delta õppehoonet kuvama. Käesoleva töö olemasolevate tööde analüüsis jõuti järeldusele, et töö autori kogemuste ja töö skoobi põhjal luuakse õppehoone korruste plaanidest 2D-s pealtvaate kuvandid.

Otsustati, et kuvandid kasutaksid skaleeruva vektorgraafika tehnoloogiat (ingl *Scalable Vector Graphics*) (edaspidi SVG). Erinevalt rastergraafikast, kus kuvand koosneb maatriksis asuvatest pikslitest [25], koosneb SVG kuvand koordinaadiandmetest [26]. Koordinaadiandmed muutuvad vastavalt kuvadi suurusele, mistõttu säilib SVG kuvandi suurendamisel kuvandi kvaliteet erinevalt rastergraafika kuvanditest, kus kuvandi suurendamisel langeb kuvandi kvaliteet [27]. Joonisel 12 on kujutatud kvaliteedi muutumist kuvandi suurendamisel.



Joonis 12. Pildikuvandite suurendamine. Vasakul on SVG kuvand, paremal on PNG kuvand.

SVG kuvandi saab lisada HTML koodi sisse, mis võimaldab kuvandile interaktiivsuse lisamist. Näiteks saab kuvanditele lisada sündmuste kuulajaid erinevate tegevuste jaoks ning korraldada kuvandi alamelementide omaduste muutmist dünaamiliselt. Üks SVG kuvandi elementidest on `<path>` element, mis võimaldab luua erinevaid kujundeid, kasutades sirg- ja kõverjooni [28]. Ruumide visualiseerimiseks saab kasutada interaktiivseid `<path>` elemente. Näiteks saab `<path>` elemendi peale vajutades kuvada lisainformatsiooni või dünaamiliselt muuta elemendi värvi vastavalt filtreeringule, et kasutajale saaks visuaalse assotsiatsiooni abil visualiseerida andmeid selgemalt. Eelnevalt kirjeldatud omadused võimaldavad luua interaktiivse majaplaani, mis säilitab kuvandi suurendamisel kvaliteeti. Olemasolevad Delta õppehoone korruste plaanid on kas rastergraafika formaadis või SVG formaadi puhul liiga detailsed, mistõttu otsustati luua uued kuvandid, mis kujutaksid ainult vajaliku informatsiooni, näiteks ruumide paigutust ja üldteenuste tingmärke.

Vaja oli valida joonistusprogramm, mis toetab vektorgraafikat. Joonistusprogrammi valimisel oli oluline, et programm toetaks järgmisi funktsionaalsusi:

- Haakumise funktsionaalsus elementide vahel. Haakumine võimaldab elemente üksteise külge ühendada nii, et elementide vahele ei jääks tühjust ning ühendatud elementide servad oleksid ühel joonel.
- Kihtide funktsionaalsus projektis. SVG kujutised luuakse Delta õppehoone esimese ja teise korruse plaanide järgi. Rastergraafika formaadis korruste plaanid saadi A. Slavini 2022. aasta bakalaureusetöö raames valminud tarkvara avalikust koodi repositooriumist⁸. Programmis peab saama importida rasterpildi formaadis korruse plaan uuele kihile, millel on muudatuste tegemine lukustatud. Korruse plaani kasutatakse kuvandi põhjana, mille järgi lisatakse kuvandile seina- ja ruumi elemendid. Muudatuste tegemise lukustamine kindlustab, et korruse plaani kogemata paigast ei liigutata.

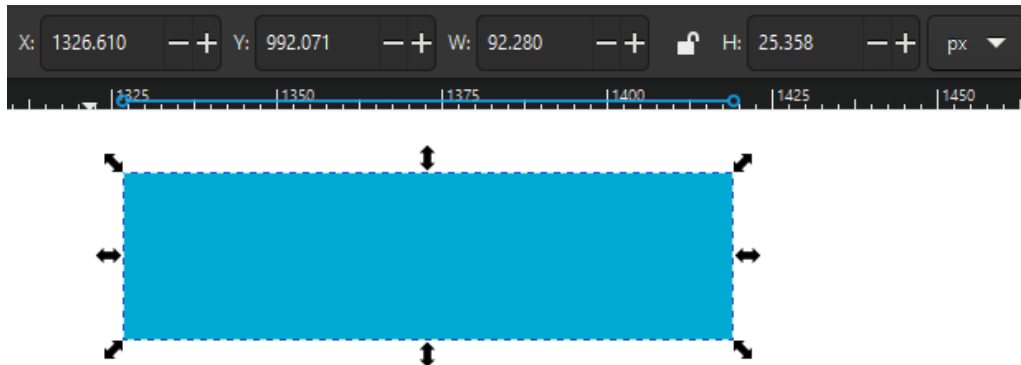
Kuvandite loomiseks valiti joonistusprogramm Inkscape⁹. Inkscape on vabavaraline ja avatud lähtekoodiga joonistusprogramm, mis toetab kirjeldatud funktsionaalsusi.

Kõigepealt loodi lihtsustatud kuvand, millel on kujutatud paari ruumiga hoonet. Nii oli võimalik töö autoril kõigepealt tutvuda Inkscape seadistuste ja tööriistadega ning teha arendusprotsessi alguses kindlaks, et SVG kuvanditega saab luua interaktiivset majaplaani. Seinade loomiseks kasutati ristküliku tööriista (ingl *Rectangle Tool*), millega saab luua

⁸ <https://github.com/tonysln/delta-wifi-pos>

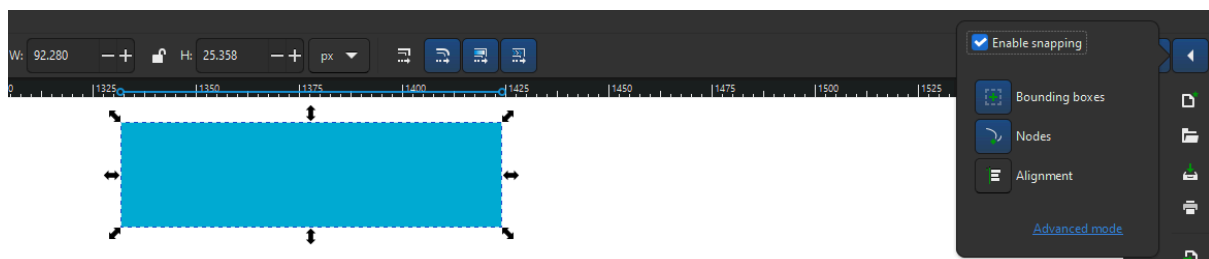
⁹ <https://inkscape.org/>

ristküliku elemente. Elemendi mõõtmeid saab valitud elemendil muuta ülemisel tööriistaribal mõõtmete väärtusi muutes või elemendi ümber olevate noolte abil (vt joonis 13).



Joonis 13. Elemendi mõõtmete muutmise võimalused. Laius (W) ja kõrgus (H) asuvad tööriista ribal.

Kirjeldatud viisidel saab lihtsalt seinte mõõtmed määrata vastavalt korruse plaanile. Elementide pööramise teeb lihtsaks ülemisel tööriistaribal asuv *Object Rotate 90* tööriist, millega saab elementi pöörata 90 kraadi vastu- ja päripäeva. Elementide haakumise saab sisse lülitada ülemisel tööriistaribal avanevas menüüs *Enable Snapping* suvandiga (vt joonis 14).



Joonis 14. Elementide haakumise menüü.

Lisaks saab hüpikaknas valida, milliste elemendi osade põhjal haagitakse. Pärast SVG kuvandile seinte paigutamist oli vaja luua iga ruumi kohta elemendid, millele saaks lisada interaktiivsust. Ruumide pindala hõlmavate elementide loomiseks kasutati sulepea tööriista (ingl *Pen Tool*). Sulepea tööriistaga saab joonestada `<path>` elemente, mis võivad olla hulknurksed ning mille nurgad ei pea olema täisnurgad. Kasutades haakumist koos sulepea tööriistaga, saab ruumi elementide nurgad ühendada ümbritsevate seinte elementidega nii, et elementide vahele ei jääks tühjust. Valminud SVG kuvandi eksportimiseks saab valida *Inkscape SVG* ja *Plain SVG* formaadi vahel. Kuvandi veebilehel kuvamiseks on .svg failist vaja kopeerida ainult SVG kuvandi kood, mille süntaks on koodis `<svg>`. Faili

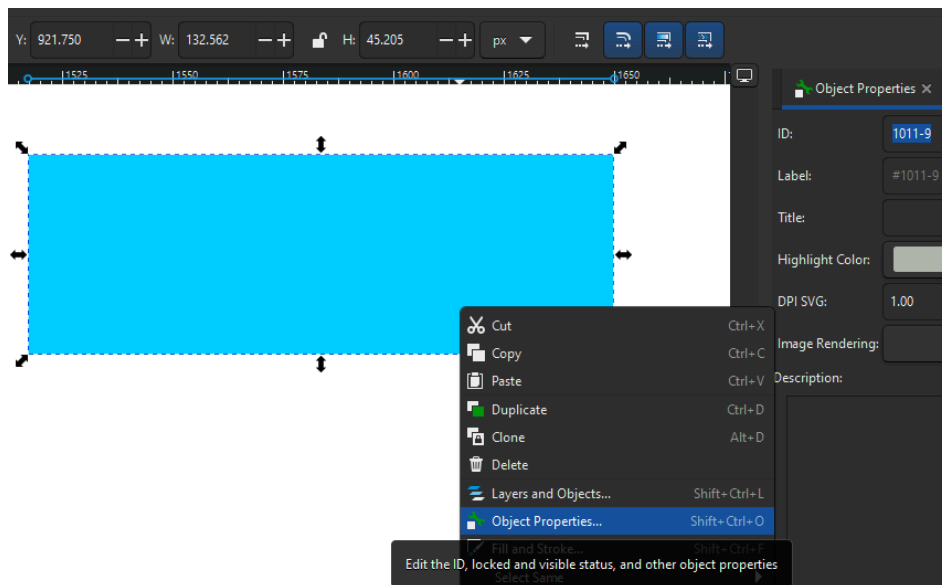
eksportimiseks valiti *Plain SVG* formaat, kuna *Inkscape SVG* formaat sisaldab kuvandi koodile lisaks Inkscape seadistuste informatsiooni, mis on kuvandi veebilehel kuvamiseks ebavajalik.

Peale kuvandi loomist peab kuvandi koodi kopeerima eksporditud .svg failist veebirakenduse koodi. Kuvandi koodile tehti muudatusi. Eemaldati koodiread, mis ei muuda kuvandi omadusi ega selle kuvamist veebilehel (vt joonis 15).

```
<path
  style="display:inline;fill:#321919;fill-opacity
  d="m 1499.5355,288.55341 h 235.0554 v 304.82501
  id="1018"
  transform="translate(1.3187499,3.7422298)" />
<path onClick={() => toggleSelectedPath("1018")}
  style={{ fill: colorValues["1018"] }}
  d="m 1505.0632,292.32884 h 235.0554 v 304.8250
  id="1018" />
```

Joonis 15. Ebavajalike koodiridade eemaldamine. Ülemine koodi näide kujutab eksporditud koodi, alumine näide kujutab sama näite muudetud versiooni veebirakenduse koodis.

Ebavajaliku koodi eemaldamine tõstab koodi loetavust ning ennetab vajadust, et arendajad peaksid kirjutatud koodi osade vajalikkust uuesti hindama. Lisaks muutub kood lühemaks, tänu millele on kuvandite mälu kasutus madalam. Järgmisena muudeti kuvandi elementide “id” väärtusi. JavaScriptis saab elemente “id” väärtuste järgi pärida, mis võimaldab mujal koodis või teistes failides muuta elementide omadusi ja lisada elementidele interaktiivsust. Elementide stiili omadusi saab seadistada *Cascading Style Sheets* (edaspidi CSS) keelega, määrates klassi- või “id” väärtuse kohta stiili seadistused .css failis. “Id” väärtusi saab muuta ka Inkscape programmis, vajutades paremat hiirenuppu elemendi peal ja valides hüpikaknast “*Object Properties*”. Seejärel kuvatakse programmi paremas küljes valitud elemendi omadusi (vt joonis 16).

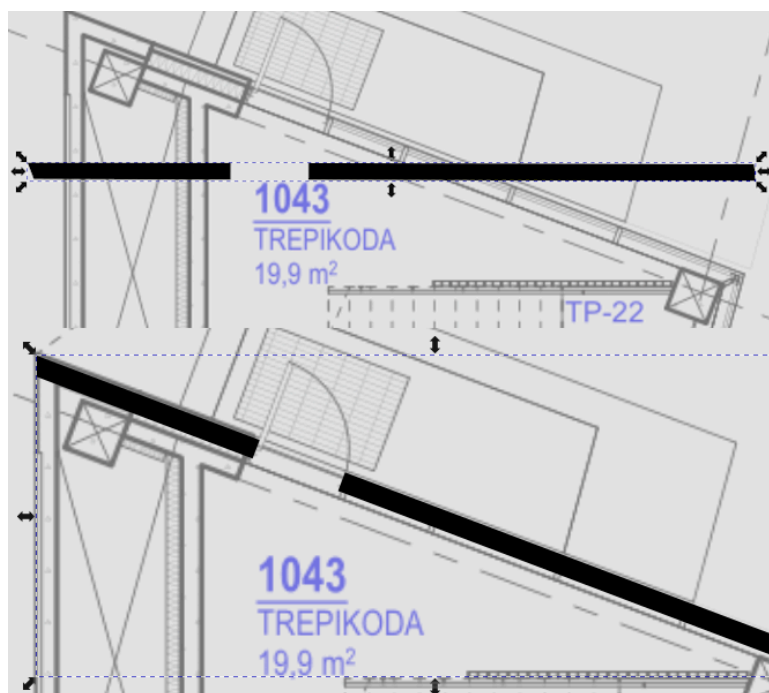


Joonis 16. Elemendi omaduste paneel.

Hiljem Delta õppehoone esimese ja teise korruste plaani kuvandite loomisel määrati elementide “id” väärtused juba Inksapes, kuna see on koodi käsitsi muutmisest kiirem meetod.

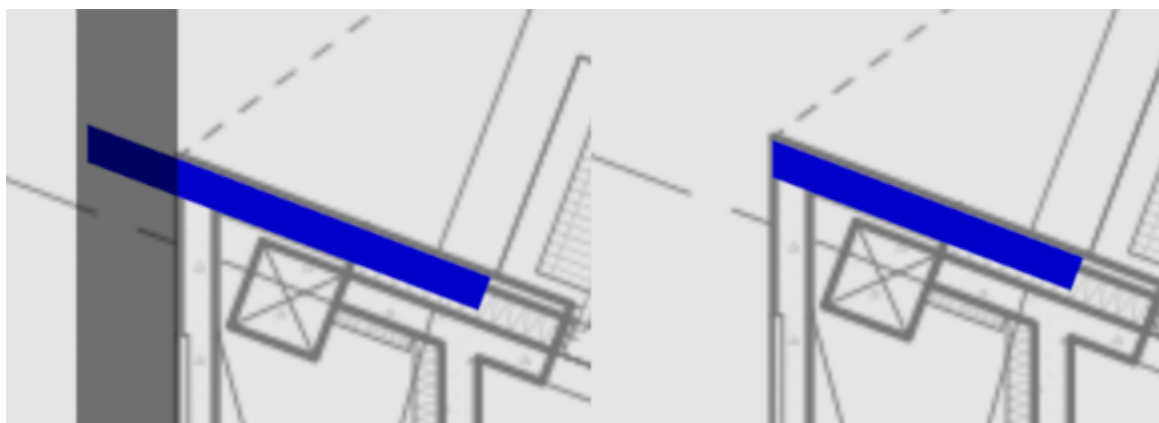
Kui oli kindlaks tehtud, et SVG kuvanditega saab luua interaktiivset majaplaani, alustati korruste plaanide kuvandite loomistega. Selle jaoks oli vaja hankida Delta õppehoone esimese ja teise korruse plaanid, mille järgi saaks luua sobilikud SVG kuvandid. Korruste kuvandite loomiseks oli lisaks eelnevalt kirjeldatud tööriistadele ja joonestamise võtetele vaja kasutada lisa tööriistu.

Korruste plaanidel ei ole kõik seinad 90° nurkade all. Seinade elementide paigutamiseks vastavalt plaanile kasutati elementide pööramist (ingl *Rotate*). Elementi saab pöörata valitud elemendi ümber olevate noolte abil või numbrilise väärtusega *Transform* paneeli *Rotate* vaheaknast. Joonisel 17 kujutatakse elemendi pööramist.



Joonis 17. Elemendi pööramine vastavalt plaanile.

Inkscape käsitleb <path> elemente ristkülikutena. Seetõttu ei saa <path> elementide mõõtmeid täpselt muuta, kui need ei ole 90° nurga all. Seina elementide mõõtmete vastavalt plaanile muutmiseks kasutati kahe <path> elemendi vahe operatsiooni (ingl *Difference*) (vt joonis 18).



Joonis 18. <path> elementide lõikamine vahe operatsiooniga.

Korruse plaani järgi muudeti uste asukohtades seinade elemendid peenemaks ning kuvadile lisati üldteenuste asukohtadele tingmärgid, et kasutajatel oleks veebirakenduses lihtsam navigeerida.

3.2 Veebirakenduse funktsionaalsus ja interaktiivsus

Veebirakenduse vajalikud funktsionaalsused kaardistati nõuete kogumise ja analüüsi käigus. Veebirakenduse funktsionaalsuste arendamiseks valiti JavaScripti programmeerimiskeel. Enamik veebilehti kasutavad kliendipoolsete toimingute haldamiseks JavaScripti [29]. Levinud tehnoloogiate kasutamine aitab kaasa platvormi jätkusuutlikkusele, kuna nii on suurem tõenäosus, et platvormi uutel arendajatel on juba eelnevad kogemused kasutatud tehnoloogiatega. Lisaks oli töö autoril olemas eelnev kogemus JavaScriptiga, mistõttu kulus tehnoloogiate õppimise peale vähem aega. Arendusprotsessi esimeses pooles ei kasutatud funktsionaalsuste arendamiseks JavaScripti raamistikku, kuid arendamise käigus jõuti järeldusele, et platvormi jätkusuutlikkuse huvides on raamistikku kasutamine mõistlikum. Veebirakenduse arendamiseks valiti Next.js raamistik järgmistel põhjustel:

- Raamistiku kasutamine võimaldab suurtes projektides tõsta koodi loetavust ja paremini organiseerida struktuuri. Näiteks saab Next.js raamistikus kasutada olekuid (ingl *state*) ja komponente (ingl *component*). Next.js raamistikus koosnevad vahelehed komponentidest ehk HTML elementidest, millele on juurde lisatud funktsionaalsust. Veebilehe koodis saab deklareerida olekuid, mida saab edastada veebilehe koodis imporditud komponentidele. Komponentid saavad neile edastatud olekute väärtusi muuta ning vastavalt oleku väärtustele saab muuta komponentide seadistust ja omadusi. Ainult JavaScripti kasutades peaks igas funktsioonis soovitud elementide seadistuste muutmiseks vastavaid elemente pärima, mistõttu oleks elementide funktsionaalsuste koodijupid omavahel segamini. Selle asemel saab kasutada tsentraalseid olekuid, mille väärtuste järgi on komponentide toimimine seadistatud. Komponente saab kasutada mitmes vahelehes korraga, mis vähendab koodi duplitseerimist.
- Next.js kasutab Node.js käituskeskkonda (ingl *runtime environment*), et hoida veebilehte kasutajatele kättesaadavana. Node.js võimaldab projekti lisateekide mugavat haldamist. Lisateekide käsitsi lisamise asemel saab Node.js abil lisateeke lisada, eemaldada ja uuendada käsurea kaudu. Lisaks tagab Node.js, et lisateegid ühilduksid korrektselt ehk lisateekide vahel ei tekiks programmi käitusajal konflikte.

3.2.1 Kuvandi liigutamine ja suurendamine

Veebirakenduses on kuvandi liigutamise- ja suumimise funktsionaalsus. Kasutajad saavad kuvandeid brauseriaknas liigutada ja suurendada vastavalt soovile, mis võimaldab kasutajatel

korruse plaani ning visualiseeritavaid andmeid detailsemalt analüüsida. Funktsionaalsus tagab kasutajasõbraliku kogemuse ka väiksema ekraaniga seadmetel, näiteks sülearvutitel ja nutitelefonidel, kus kasutaja ei pruugi algse suurusega kuvandilt informatsiooni välja lugeda.

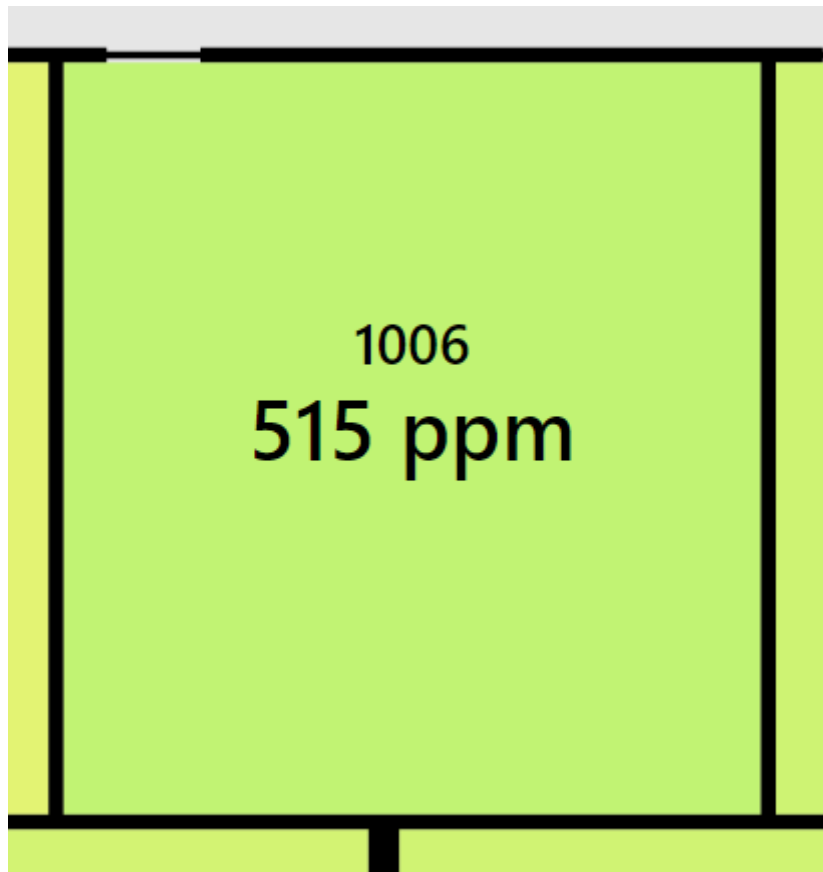
Kõigepealt prooviti funktsionaalsust arendada ilma lisateekiteta, kuna lisateekide kasutamine suurendab platvormi sõltuvust teistest tehnoloogiatest. Lisateegi arendamise lõpetamine võib muuta platvormi kasutuskõlbmatuks. Funktsionaalsuse arendamiseks prooviti kasutada sündmuste kuulamisi ja matemaatilisi valemeid. Sündmused on kasutaja tegevused brauseriaknas, näiteks hiire rulliku kerimine või hiire nuppude vajutamine. Valemid koosnesid vastavalt operatsioonile parameetritest. Näiteks kasutati kuvandi- ja brauseriakna mõõtmeid ning kuvandi ja hiire koordinaate brauseriaknas. Kirjeldatud väärtusi kasutati valemites, millega arvutati kuvandi suurus või asukoht brauseriaknas. Funktsionaalsuse arendamisel ja testimisel tulid välja mitmed piirijuhtumid (ingl *edge case*). Piirijuhtumite parandamiseks oli vaja luua arvukaid lisakontrolle, mis muutsid koodi järjest keerulisemaks. Koodi kirjutamine lisateekideta osutus keeruliseks ja veaohlikus, mistõttu jõudis töö autor järeldusele, et mõistlikum on kasutada funktsionaalsust pakkuvat lisateeki.

Lisateegiks valiti D3.js, milles on SVG kuvandi liigutamise- ja suumimise funktsionaalsus sisse ehitatud. npm registri D3.js lisateegi leheküljelt¹⁰ on näha, et lisateek on populaarne, seega lisateegi arendamise lõpetamise oht on madal. Tänu lisateegile on funktsionaalsuse kood arusaadav ja töökindel. Lisaks võimaldab D3.js luua sujuvaid kuvandi liigutamise ja suumimise animatsioone.

3.2.2 Andmete visualiseerimine kuvandil

Veebirakenduses visualiseeritakse temperatuuri, CO2 kontsentratsiooni ja ajaplaani andmeid. Iga ruumi kohta on kuvandil ruumi pindala hõlmav `<path>` element. Joonis 19 kujutab andmete visualiseerimist `<path>` elemendil.

¹⁰ <https://www.npmjs.com/package/d3>



Joonis 19. Andmete visualiseerimine <path> elemendil.

<path> elemendi peal visualiseeritakse ruumi numbrit ja filtreeritud andmetüübi väärtust tekstina. Kuvandite komponentides luuakse tsüklite abil iga ruumi kohta <text> elemendid, mis seatakse vastava <path> elemendi koordinaatide põhjal ruumi keskele. Kirjeldatud paigutus võimaldab kasutajal lihtsalt aru saada, mis on uuritava ruumi number ja filtreeritud andmetüübi väärtus. Ruumile vastava <path> elemendi taustavärv tuleneb filtreeritud andmetüübi väärtusest. Värvid võimaldavad kasutajal visuaalse assotsiatsiooni abil lihtsamini aru saada, kas väärtus on näiteks liiga kõrge või madal. Andmete värskendamisel uuendatakse dünaamiliselt <text> elementide väärtused ja <path> elementide taustavärvid.

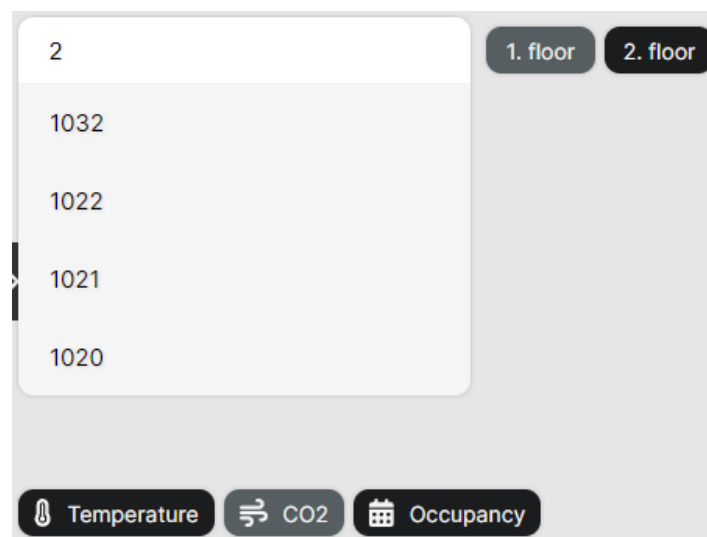
Pidevate väärtuste puhul kasutatakse värvi interpoleerimist. CO₂ kontsentratsiooni puhul eeldatakse, et väärtused jäävad 300 ppm ja 1000 ppm vahemikku. Allen jt [30] on leidnud, et inimestel on 1000 ppm kontsentratsiooniga keskkonnas 15% madalam kognitiivne võimekus võrreldes keskkonnaga, kus kontsentratsioon on ~500 ppm ringis. 300 ppm ja väiksema väärtuse kohta on määratud roheline värv ning 1000 ppm ja suurema väärtuse kohta on määratud punane värv. Vahemiku vahepealsete väärtuste kohta interpoleeritakse värv vastavalt rohelise ja punase värvi vahelt. 650 ppm korral on interpolatsiooni tulemusena

taustavärv kollane. Temperatuuri väärtuste puhul on kaks vahemikku. 18°C kuni 22°C vahemikus on värvid vastavalt helesinine ja roheline ning 22°C kuni 26°C vahemikus roheline ja punane.

Diskreetsete väärtuste puhul kasutatakse kindlaid värve. Ajaplaani puhul on ruumi värvus punane, kui ruum on reaalajas ajaplaani järgi reserveeritud. Reserveeringu puudumisel on ruumi värvus roheline.

3.2.3 Andmete ja kuvandite filtreerimine

Veebirakenduses saab korruse plaani ja andmetüüpi filtreerida nuppude kaudu. Joonis 20 kujutab korruse ja andmetüübi filtreerimise nuppe.



Joonis 20. Otsinguriba ja filtreeringute nupud.

Nupud on alati nähtaval, et kasutajad ei peaks filtrite muutmiseks teostama lisategevusi. Korruse filtreerimise nupud asuvad otsinguriba kõrval, brauseriakna ülemises vasakus nurgas. Andmetüüpide filtreerimise nupud asuvad brauseriakna alumises vasakus nurgas. Vajutades nupule, muudetakse vastavalt nupu väärtusele visualiseeritavat korruse plaani või andmetüüpi. Aktiivsele filtrile vastava nupu värvus on heledam, et kasutajal oleks lihtne eristada aktiivseid filtreid. Väikese ekraani formaadis asuvad korruse ja andmetüübi filtreerimise nupud veebirakenduse üleval osas, otsinguriba all.

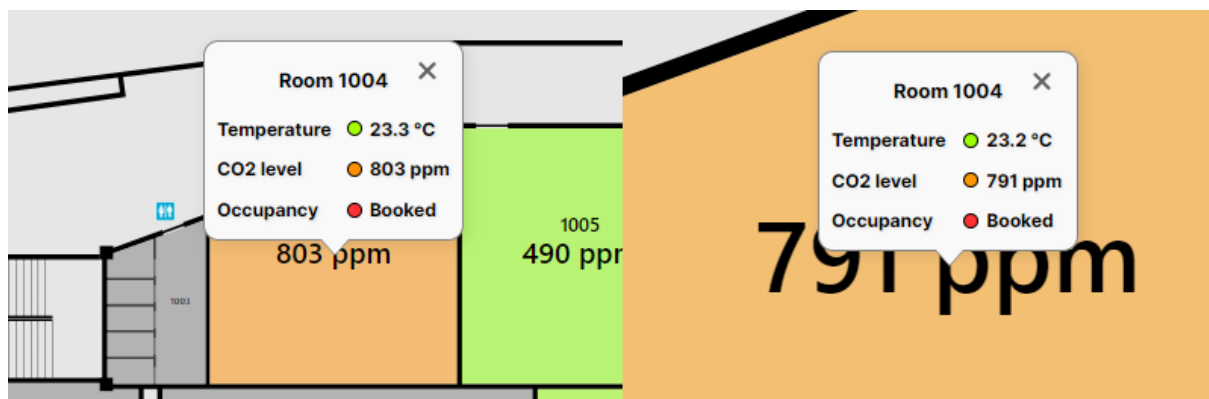
3.2.4 Otsinguriba

Veebirakenduses saab korruse plaanil ruumide asukohti üles leida otsinguriba abil. Sisestades otsinguribale ruumi numbri, kuvatakse kasutajale soovitusi, mis sisaldavad otsinguribale sisestatud väärtust (vt joonis 20). Ruumi leidmiseks saab pakutud soovitustele peale hiirega

vajutada. Valides ruumi numbri, muudetakse korruse filter vastavalt ruumi asukohale, tekitatakse ruumi keskele hüpikaken ning liigutatakse sujuva animatsiooniga korruse plaani nii, et valitud ruumi <path> element oleks brauseri akna keskel. Hüpikaken ja sujuv animatsioon võimaldab kasutajal lihtsalt aru saada, kus on valitud ruumi asukoht. Väikese ekraani formaadis asub otsinguriba veebirakenduse üleval osas.

3.2.5 Hüpikaken

Veebirakenduses tekitatakse valitud ruumi keskele hüpikaken. Hüpikaken aitab kasutajal leida valitud ruumi asukohta. Hüpikaknas kuvatakse valitud ruumi numbrit ning ruumi kõiki reaalaja andmeid, et kasutaja saaks valitud ruumi kõiki andmeid korraga näha ühest kohast. Joonis 21 kujutab hüpikakent, mis kuvab ruumi kõiki andmeid kompaktselt.

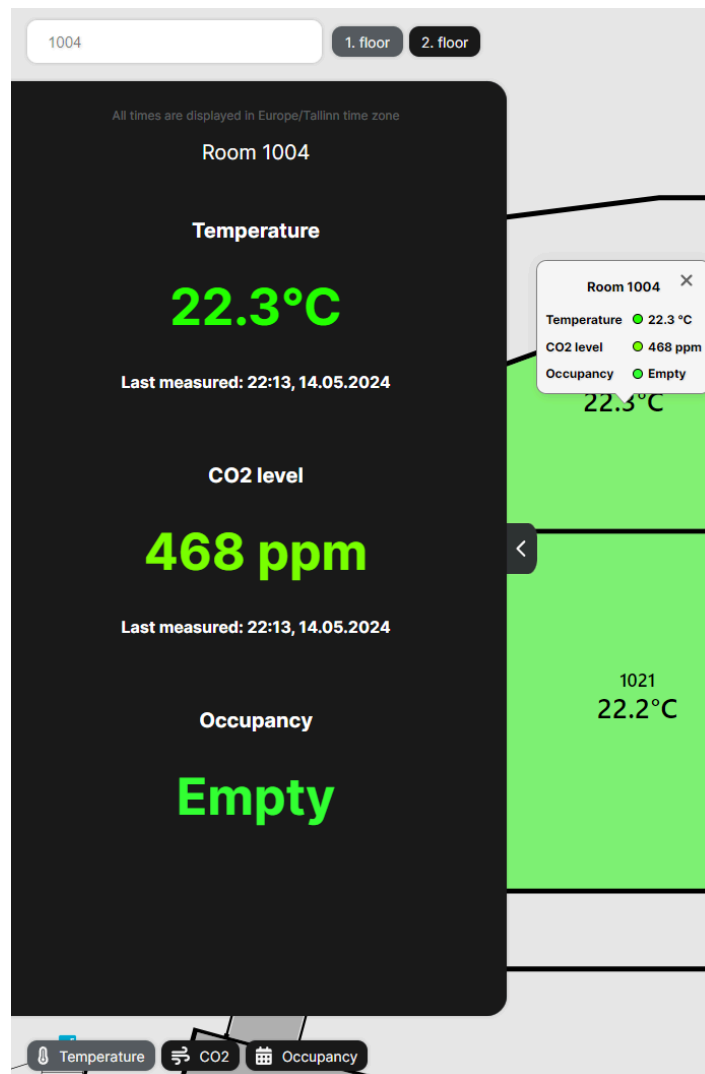


Joonis 21. Hüpikaken, mis säilitab suuruse sõltumata kuvandi suumi tasemest.

Lisaks on jooniselt näha hüpikakna suuruse sõltumatus kuvandi suurenduse tasemest, et hüpikaknas olev informatsioon oleks loetav ka väiksema suurenduse korral ning vältida kasutaja navigeerimise takistamist suurema suurenduse korral.

3.2.6 Külgpaneel

Veebirakenduses saab valitud ruumi andmete kohta detailsema ülevaate brauseri akna vasakul küljel asuvalt paneelilt. Joonis 22 kujutab laiendatud külgpaneeli, kus on visualiseeritud valitud ruumi andmed.

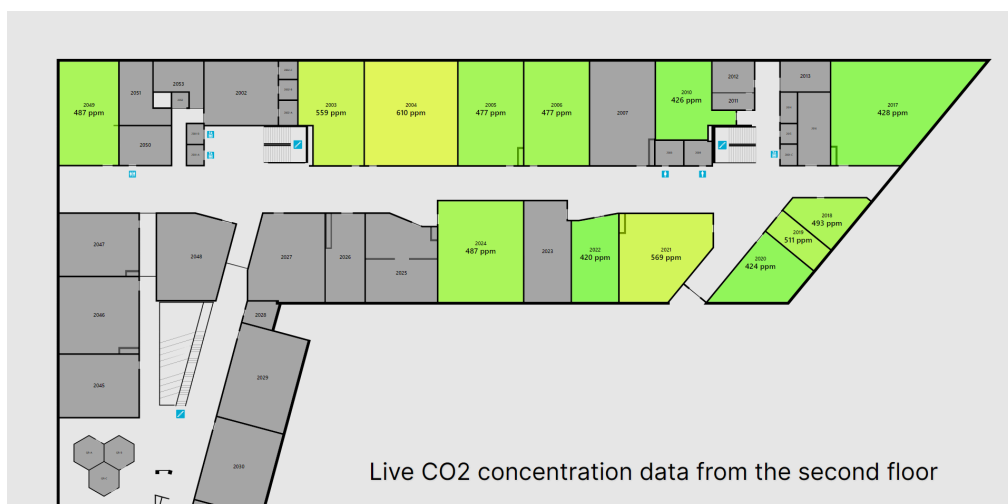


Joonis 22. Laiendatud külgpaneel.

Lisaks andmetüüpide väärtusele on välja toodud väärtuse mõõtmise kellaaeg ning reserveeringu puhul nimetus, tüüp ja ajavahemik. Veebilehe esmasel laadimisel on külgpaneel peidetud, paneeli nähtavust saab kasutaja muuta küljepaneeli nupule vajutades. Väikese ekraani formaadis asub paneel veebirakenduse alumises osas.

3.2.7 Visualisatsioon ekraanile

Veebirakendusega saab tekitada visualisatsiooni, milles korrust ja andmetüüpi filtreeritakse internetiaadressi kaudu. Joonis 23 kujutab internetiaadressi kaudu seadistatud visualisatsiooni.



Joonis 23. Internetiaadressi kaudu seadistatud visualisatsioon.

Visualisatsioone saab tekitada veebirakenduse aadressi `/display` otspunkti kaudu, millele on juurde vaja lisada lisaparaameetrid kujul `"/display?floor=soovitud_korrus&filter=soovitud_andmetüüp"`. Korruse ja andmetüübi valimiseks peab paraemetritele "floor" ja "filter" määrama soovitud filtreeringute väärtused (vt tabel 2).

Paraameeter	Kirjeldus	Sobivad väärtused
floor	Määrab kuvatava korruse plaani	first, second
filter	Määrab kuvatavate andmete tüübi	temperature, co2, occupancy

Tabel 2. Veebirakenduse `/display` otspunkti paraameetrid.

Visualisatsioonis on korruse plaan suurendatud kujul, mida liigutatakse sujuva animatsiooniga alumisest servast ülemise servani. Sisse suunitud kuvandi liigutamisel on võimalik kasutajal näha kõikide ruumide andmeid nii, et need oleks selgelt välja loetavad. Lisaks kuvatakse visualisatsiooni alumises osas filtreeritava korruse ja andmetüübi nimetust.

3.3 Andmete pärimine ja edastamine

Veebirakenduses visualiseeritavaid andmeid hoiustatakse platvormi välistes teenustes. Andmete kättesaamiseks peab pärima vastavaid teenuseid. Temperatuuri ja CO2 andmeid hoiustatakse Delta keskuse Cumulocity keskkonnas ja InfluxDB andmebaasis. TÕ hoonete ajaplaanide andmeid hoiustatakse ÕIS 2 platvormis.

3.3.1 Temperatuuri ja CO2 andmed

Temperatuuri ja CO2 andmeid töötleb eraldiseisev Node.js teenus, mille kood asub “queryCumulocity.js” failis. Temperatuuri ja CO2 andmed päritakse Delta keskuse Cumulocity keskkonnast. Algselt plaaniti andmeid pärida InfluxDB andmebaasist. Arendusprotsessi käigus selgus, et Cumulocity keskkonnast ei edastata kõiki andmeid InfluxDB andmebaasi, mistõttu otsustati andmeid pärida Cumulocity API kaudu. Reaalaja andmete kättesaamiseks saab Cumulocity API kaudu pärida kindlas ajavahemikus seadme poolt mõõdetud andmeid.

Delta keskuse Cumulocity keskkonnale on ligipääs piiratud. Andmete pärimise otstarbeks loodi Cumulocity keskkonnas eraldi kasutaja, millele on antud ainult lugemispääs seadmete seadistuste ja seadmete mõõdetud andmete üle. Kasutaja informatsioon lisatakse päringu päise “Authorization” parameetrina.

Pärimise loogikas luuakse päringud mitmete seadmete kohta. Ruumi numbrid viidi seadmetega vastavusse failis “room_mapping.json”, mis on JSON¹¹ formaadis. Kirjeldatud fail võimaldab ruumi numbri ja seadmete vahelisi seadistusi visualiseerida ning muuta kiirelt ja mugavalt tsentraalsest kohast. Päringud luuakse ainult siis, kui failis on ruumi numbri kohta olemas “qe_device_id” väärtus ning mõne andmetüübi väärtus, mille kohta kogutakse Cumulocity keskkonnas seadme poolt andmeid (vt joonis 24).

```
"1020": {  
  "has_events": true,  
  "description": "Auditoorium",  
  "qe_device_name": "AK01'Vent'SV2'Vav3",  
  
  "qe_device_id": "176",  
  "qe_temperature": "T Su2",  
  "qe_co2": "DP"  
},
```

Joonis 24. Ruumi 1020 kirje failis “room_mapping.json”.

Päring loetakse edukas, kui päringu vastuse staatuse kood on 200 [31]. Kui eduka päringu vastus sisaldab etteantud ajavahemikus mõõdetud andmeid, siis võetakse andmetest esimene andmepunkt ja salvestatakse JSON formaadis faili “room_measurements.json”. (vt joonis 25).

¹¹ <https://www.json.org/json-en.html>

```

"1020": {
  "qe_temperature": "21.7",
  "qe_temperature_time": "2024-05-01T08:14:00.000Z",
  "qe_co2": "466",
  "qe_co2_time": "2024-05-01T08:14:00.000Z"
},
"1021": {
  "qe_temperature": "22.3",
  "qe_temperature_time": "2024-05-01T08:14:00.000Z",
  "qe_co2": "No data",
  "qe_co2_time": "No data"
},

```

Joonis 25. Kirjed failis “room_measurements.json”.

Delta õppehoones mõõdavad seadmed andmeid erinevate intervallidega, näiteks osa seadmetest mõõdavad iga kolme sekundi tagant ning osa seadmetest iga minuti tagant. Päritud ajavahemikust esimese andmepunkti valimine ühtlustab ajahetke, millal andmeid mõõdeti. Temperatuuri ja CO2 andmeid päritakse iga minuti tagant, et tagada andmete võimalikult reaalaajale lähedast täpsust.

3.3.2 Ajaplaanide andmed

Ajaplaanide andmeid töötleb eraldiseisev Node.js teenus, mille kood asub “queryTimetables.js” failis. Delta õppehoone esimese ja teise korruse ruumide ajaplaanid päritakse ÕIS 2 keskkonnast. ÕIS 2 keskkonnale on loodud API¹², millele on ligipääs avalik. Ruumi ajaplaani saab pärida “https://ois2.ut.ee/api/timetable/room” otspunkti kaudu. Erinevalt Cumulocity API päringust, kus parameetrite väärtused asuvad internetiaadressil, lisatakse ÕIS 2 API päringu parameetrite väärtused päringu sisuna kaasa. Joonis 26 kujutab ruumi ajaplaani päringu sisu struktuuri.

```

{
  "building": "string",
  "room": "string",
  "date": "string"
}

```

Joonis 26. /api/timetable/room otspunkti päringu sisu struktuur¹³.

Päringus kasutatakse kolme parameetrit:

¹² <https://ois2.ut.ee/docs/>

¹³

<https://ois2.ut.ee/doc/timetable/swagger-ui/index.html?url=/doc/timetable/api-docs#/timetable-controller/fetchRoomTimetable>

- “building” - hoone väärtus ÕIS 2 keskkonnas. Delta õppehoonele vastab väärtus "NAR18OH"¹⁴.
- “room” - ruumi väärtus ÕIS 2 keskkonnas.
- “date” - kuupäev, formaadis YYYY-MM-DD.

Päringud luuakse vastavalt “room_mapping.json” failis ruumi numbrile vastava “has_events” parameetri väärtusele, et vältida ebavajalike päringute tegemist. Eeldatakse, et teatud ruumide jaoks ei tehta ÕIS 2 keskkonnas reserveeringuid, nagu näiteks tualettruumid ja trepikojad.

Lisaks asuvad Delta õppehoone teisel korrusel õpilastele mõeldud koostööruumid, mida saavad õpilased broneerida DeltaQR platvormi kaudu. Koostööruumide ajaplaanidele on ligipääs avalik. Veebibrauseri tööriistade abil uuriti DeltaQR veebilehel võrguliiklust, et leida pärimise loogika jaoks DeltaQR API ajaplaanide otspunktid.

Kui päring on edukas ning vastus sisaldab etteantud kuupäeval olemasolevaid reserveeringuid, salvestatakse reserveeringud omakorda JSON formaadis faili “room_timetables.json” (vt joonis 27).

```

"1020": {
  "current_event": null,
  "events": [
    {
      "title": "Matemaatiline maailmapilt",
      "study_work_type": "practice",
      "begin_time": "10:15",
      "end_time": "12:00"
    },
    {
      "title": "Funktsionaalanalüüs I",
      "study_work_type": "practice",
      "begin_time": "14:15",
      "end_time": "16:00"
    },
    {
      "title": "Andmeanalüüs II",
      "study_work_type": "lecture",
      "begin_time": "16:15",
      "end_time": "18:00"
    }
  ]
},

```

Joonis 27. Ruumi 1020 kirje failis “room_timetables.json”.

Eeldatakse, et päeva alguseks on ruumide reserveeringud paigas, seega ajaplaane päritakse ühe korra ööpäevas. Koostööruumide ajaplaane päritakse iga minuti tagant, kuna õpilased

¹⁴ Väärtus pärineb töö autorile ÕIS 2 toe poolt 20. märtsil 2024. aastal saadetud e-kirjast.

saavad enda broneeringuid igal ajahetkel muuta. Teenuses kontrollitakse iga minuti tagant, kas praegusel ajahetkel on ruum reserveeritud, mille põhjal muudetakse parameetri “current_event” väärtust vastavalt.

3.3.3 Andmete edastamine

Andmete edastamiseks loodi eraldiseisev API teenus. Teenuse loomiseks kasutati Express.js raamistikku, mis võimaldab API loomist kergelt arusaadava koodiga. Andmete edastamiseks loodi kaks otspunkti:

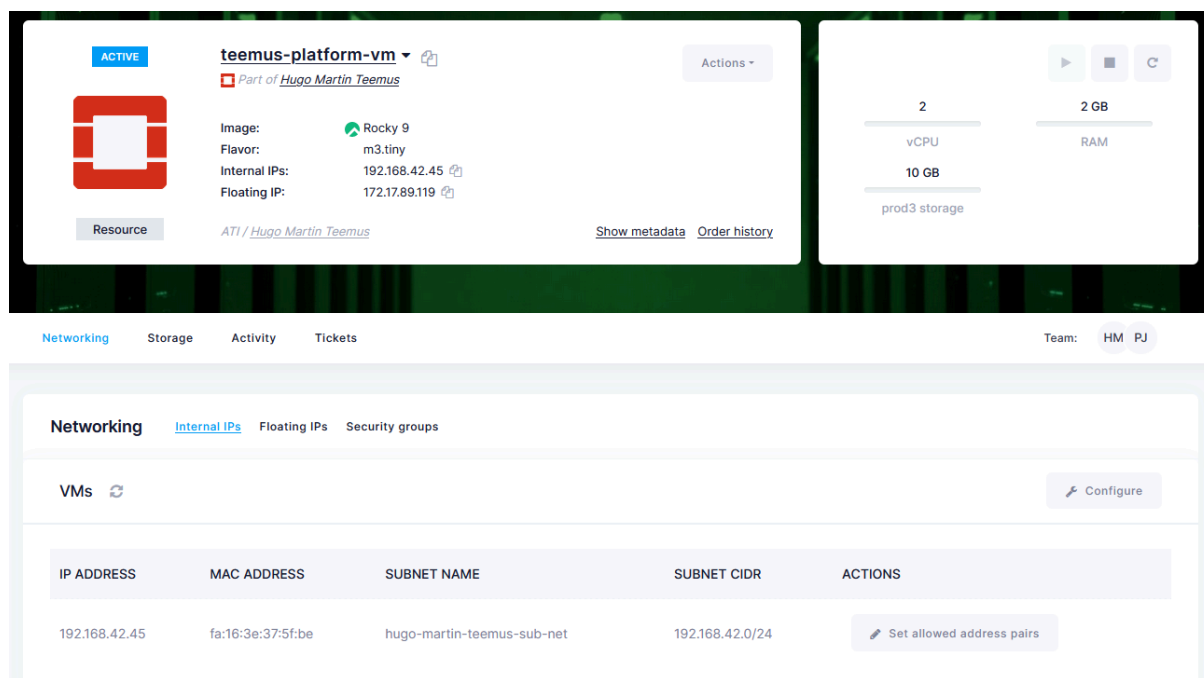
- /api/measurements - tagastab “room_measurements.json” faili sisu.
- /api/timetables - tagastab “room_timetables.json” faili sisu.

Platvormi väliseid teenuseid päritakse iga täisminuti järel. Veebirakendus pärib API kaudu andmeid iga minuti järel, 30 sekundit pärast täisminutit. Veebirakenduse poolne viivitus kindlustab, et väliste teenuste pärimise protsessid on lõpetatud ning otspunktides tagastatav andmestik on uuendatud.

3.4 Platvormi ülesseadmine

Platvormi ülesseadmiseks kasutatakse tarkvara Docker. Komponentidest moodustatakse Dockeri konteinerid (ingl *Docker container*), millega tagatakse komponentide sõltumatus konteinerit majutava seadme seadistustest ja eripäradest. Platvormi majutatakse virtuaalmasinas, mis asub ETAIS¹⁵ keskkonna TÜ High Performance Computing (edaspidi HPC) süsteemis. Virtuaalmasin kasutab Rocky Linux operatsioonisüsteemi. Platvormi majutamine ei nõua palju jõudlust, seega virtuaalmasina jaoks määrati kaks protsessori tuuma ja kaks GB RAM-i. Joonis 28 kujutab loodud virtuaalmasina juhtpaneeli ETAIS keskkonnas.

¹⁵ <https://etais.ee/>



Joonis 28. Virtuaalmasina juhtpaneel ETAIS keskkonnas.

Ülesseadmiseks luuakse kõigepealt komponendi koodi repositooriumisse Docker piltide (ingl *Docker image*) seadistusi sisaldavad failid “Dockerfile” ja “docker-compose.yaml”. “Dockerfile” failis seadistatakse teenuse ülesseadmine. “docker-compose.yaml” failis konfigureeritakse konteineri suvandid, nagu konteineri port ja ajatsoon. Konteinerid luuakse piltide põhjal. Seejärel peab kloonima platvormi koodi repositooriumid virtuaalmasina failisüsteemi. Docker konteinerite loomiseks jooksutatakse komponendi kaustas käsk “docker compose up --build”, mis tekitab seadistuste põhjal pildi, seejärel loob ja käivitab konteineri. Teenuste uuendamiseks tuleb olemasolevad pildid ja konteinerid eemaldada, uuendada kloonitud koodi vastavas asukohas “git pull” käsuga ning seejärel uuendatud koodi põhjal luua pildid, konteinerid ning loodud konteinerid käivitada.

4. Valminud platvormi analüüs

Järgnevas peatükis analüüsitakse valminud platvormi vastavust 2.2 peatükis kirjeldatud nõuetele. Lisaks kirjeldatakse platvormi võimalikke edasiarendusi.

4.1 Vastavus nõuetele

- Uues lahenduses peab saama ruumis kogutud andmeid otsida ruumi numbri ja asukoha põhjal.

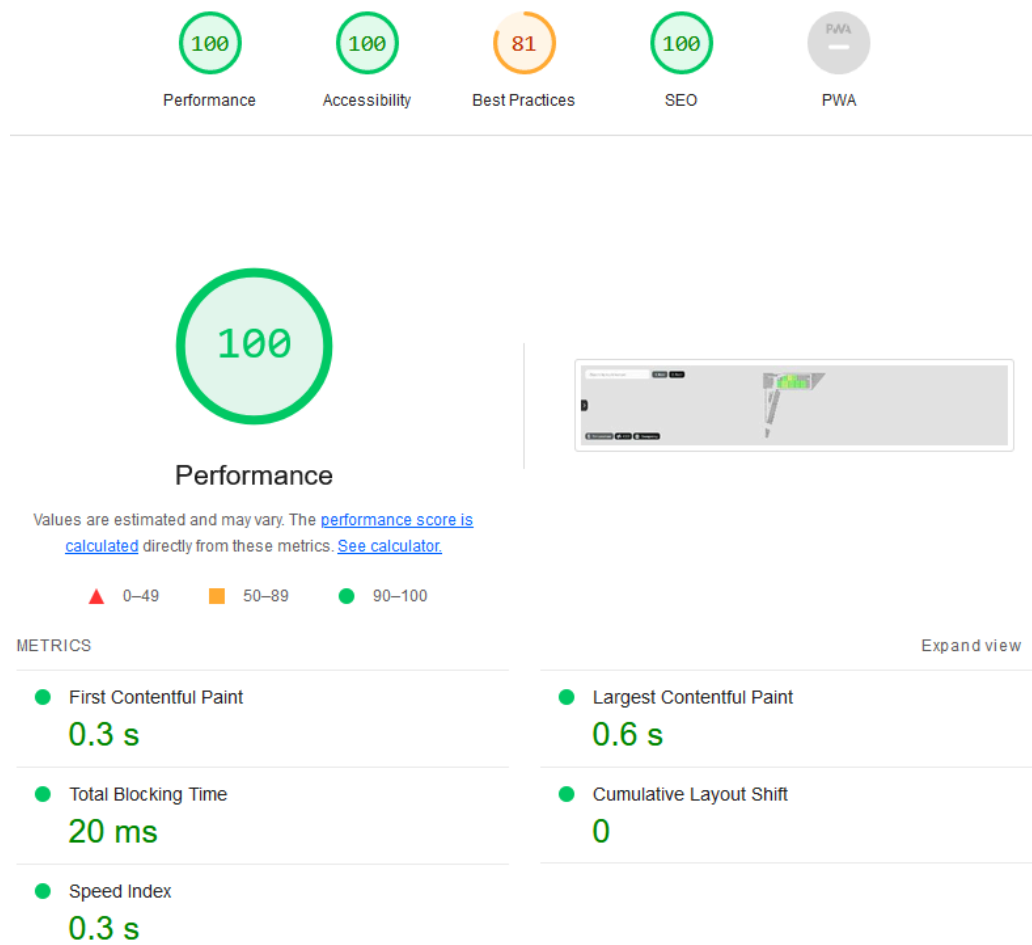
Valminud platvormi veebirakenduses visualiseeritakse andmeid korruste kuvandite peal, mis võimaldab kogutud andmeid näha ka nende asukoha põhjal, seega nõue on täidetud. Lisaks saab platvormis otsida andmeid otsinguriba kaudu ruumi numbri põhjal.

- Uus lahendus peab olema avalikult ligipääsetav ilma lisafailide allalaadimiste ja rakenduste installeerimistega.

Andmete visualiseerimise rakendus on loodud veebikeskkonda. Valminud platvormi majutatakse TÜ HPC süsteemis, mis tähendab, et veebirakendusele ligipääsemiseks on lisaks veebibrauserile vaja TÜ võrguga suhtlevat internetiühendust. TÜ võrk on ligipääsetav TÜ hoonetes kättesaadava avaliku internetivõrgu või virtuaalse privaativõrgu tunneli kaudu. Veebirakendusele pääseb ligi ilma lisafailide allalaadimiste ja rakenduste installeerimistega, seega nõue on täidetud.

- Uues lahenduses peab olema tagatud kasutajasõbralik kogemus ka nõrgema jõudlusega seadmetel.

Veebirakenduse jõudluse testimiseks kasutati Google Lighthouse tööriista. Lighthouse võimaldab lihtsalt testida veebilehe jõudlust ning testide tulemuste põhjal genereerida raporti [32]. Joonis 29 kujutab veebirakenduse kohta loodud Lighthouse raportide jõudluse näitajaid. Veebirakenduse raporti jõudluse skoor on suurepärane, seega nõue on täidetud.



Joonis 29. Veebirakenduse kohta loodud Lighthouse raporti jõudluse näitajad.

Best Practices skoor on madalam, kuna veebirakendus ei toeta HTTPS protokoll. Platvormile ei ole määratud domeeni nime. Näiteks saab Let's Encrypt tehnoloogia abil genereerida ainult domeeni nime põhiselt TLS sertifikaadi, mida on vaja HTTPS toe jaoks [33].

- Uues lahenduses peab olema funktsionaalsus tekitada visualisatsioon, milles on visualiseeritavat korrust ja andmetüüpi võimalik filtreerida internetiaadressisse sisestatud filtreeringute järgi.

Veebirakenduse aadressi “/display?floor=soovitud_korrus&filter=soovitud_filter” otspunkti kaudu saab tekitada vastavalt parameetrite väärtustele seadistatud visualisatsiooni, seega nõue on täidetud.

- Uus lahendus peab olema jätkusuutlik. See tähendab, et platvormi lähtekood on koodi repositooriumist kättesaadav, koodi on lisatud selgitavad kommentaarid ja koodi repositoorium sisaldaks platvormi töövoo vooskeemi.

Valminud platvormi komponentide lähtekoodid on avalikult kättesaadavad töö autori GitHubist. Koodi on lisatud selgitavaid kommentaare ning koodi repositooriumid sisaldavad platvormi kirjeldust, töövooskeemi ning platvormi ülesseadmise samme. Nõue on täidetud.

4.3 Edasiarenduse võimalused

Järgnevas alampeatükis tuuakse välja platvormi võimalikud edasiarendused.

- Mitmete esimese ja teise korruse ruumide kohta ei koguta Cumulocity keskkonda andmeid, mistõttu ei saa vastavate ruumide andmeid visualiseerida. Rohkemate andmete visualiseerimiseks peaks kõigepealt seadistama vastavate ruumide andmete kogumist Cumulocity keskkonnas ning seejärel korraldada ruumide ja seadmete vahelised suhted *back-end* komponendi “room_mapping.json” failis.
- Veebirakenduse küljepaneelile oleks võimalik juurde lisada visualisatsioone. API /api/timetables otspunkti kaudu edastatakse iga ruumi kohta päeva ajaplaan. Hetkel kuvatakse küljepaneelil ruumi reserveeringu staatust ainult reaalajas. Näiteks saaks ruumi reaalaja hõivatuse juurde arendada ruumi ajaplaani visualisatsiooni, kus oleks kuvatud kõik päeva reserveeringud. Lisaks saaks temperatuuri ja CO₂ väärtuse juurde lisada graafikud, mis kujutaksid näiteks viimase tunni aja andmeid. Näiteks oleks graafiku abil näha, kui kiiresti ning kui palju tõuseb CO₂ kontsentratsiooni väärtus siis, kui ruum täitub inimestega.
- Delta õppehoones asuvad kolmandal korrusel koosolekuruumid, mida saab broneerida Skedda süsteemi kaudu. Võimalikus arenduses saaks luua kolmandast korrusest lihtsustatud kuvandi ning Delta õppehoone Skedda süsteemi¹⁶ pärimise loogika *back-end* komponendile, et visualiseerida kolmanda korruse koosoleku ruumide hõivatust.
- Valminud platvormil puudub HTTPS (ingl *Hypertext Transfer Protocol Secure*) tugi. Platvorm asub TÜ HPC süsteemi virtuaalmasinas ning platvormile pääseb ligi ainult TÜ võrgus olev seade virtuaalmasinale määratud avaliku IP aadressi kaudu. Võimalik arendus oleks hankida platvormile domeeninimi ja muuta platvorm kättesaadavaks ka väljaspoolt TÜ võrku. Domeeni nimele saab Let's Encrypt tehnoloogiaga genereerida TLS (ingl *transport layer security*) sertifikaadi, et tagada turvaline ühendus HTTPS protokolliga.

¹⁶ <https://unitartucs.skedda.com/booking>

- Veebirakenduse otsinguriba soovitud kuvavad ruumi numbrit. *Back-end* komponendi “room_mapping.json” fail sisaldab igale ruumi numbrile vastava ruumi nimetust, nagu näiteks “Labor” ja “Seminariruum”. Võimalik arendus oleks *front-end* komponenti kopeerida “room_mapping.json” fail, et failist saaks pärida ruumide nimetusi ning lisada need otsinguriba soovitudes ruumi numbri kõrvale.
- Veebirakenduse funktsionaalsusi testiti manuaalselt. Näiteks kontrolliti käsitsi ruumi valimisel, et valitud ruumi keskele tekiks hüpikaken, küljepaneelil visualiseeritakse valitud ruumi informatsiooni ning otsinguribal kuvatakse valitud ruumi numbrit. Võimalik arendus oleks uurida võimalusi automaatsete testide loomiseks ning välja arendada automaatsed testid veebirakenduste funktsionaalsuste kontrollimiseks.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli uurida erinevaid lähenemisviise andmete visualiseerimiseks, analüüsida olemasolevat Delta õppehoone andmete visualiseerimise lahendust ning uurimise ja analüüsi tulemuste põhjal luua uus platvorm Delta õppehoone andmete visualiseerimiseks.

Uurimise käigus analüüsiti sama valdkonna olemasolevaid lahendusi ja uurimistöid, et kaardistada erinevate lähenemiste head küljed ja puudujäägid. Lisaks analüüsiti Delta õppehoone andmete visualiseerimise varasemat lahendust, et leida probleemkohad, mida on vaja uue platvormiga parandada. Analüüsi põhjal pandi kirja uue platvormi nõuded.

Valminud platvormis visualiseeritakse Delta õppehoone esimese ja teise korruse ruumide temperatuuri, CO₂ kontsentratsiooni ja ajaplaani andmeid. Platvorm jaguneb *front-end* ja *back-end* komponentideks.

Front-end komponent sisaldab Next.js raamistikus loodud andmete visualiseerimise veebirakendust, mis kasutab esimese ja teise korruse SVG formaadis kuvandeid. Korruse plaani kuvandid loodi Inkscape joonistusprogrammiga.

Back-end komponent sisaldab andme pärimiste teenuseid ning API teenust andmete edastamiseks veebirakendusele. Temperatuuri ja CO₂ andmed pärinevad Delta keskuse Cumulocity keskkonnast ja ajaplaani andmed ÕIS 2 ja DeltaQR keskkonnast.

Töö esimeses peatükis antakse ülevaade töö käigus käsitletud teemadest ja tehnoloogiatest ning analüüsitakse valdkonnaga seotud olemasolevaid lahendusi ja uurimistöid. Teises peatükis analüüsitakse olemasolevat lahendust ning kirjeldatakse valminud platvormi nõudeid. Kolmandas peatükis tutvustatakse valminud platvormi arhitektuuri ning kirjeldatakse platvormi lahenduskäiku. Neljandas peatükis analüüsitakse valminud platvormi vastavust nõuetele ning tuuakse välja edasiarenduste võimalused.

Valminud platvormi veebirakendus asub aadressil <http://172.17.89.119.nip.io/>. Veebirakendusele ligipääsemiseks on vaja ühendust TÜ võrguga.

Viidatud kirjandus

- [1] Tartu Ülikooli Delta Keskus. Avasta Deltat. <https://delta.ut.ee/koostoo/avasta-delta/> (29.11.2023)
- [2] EKI teatmik: Eesti õigekeelsuskäsiraamat. Euroopa Liidu sõnavõistlus 2014. 2021. <https://www.eki.ee/teatmik/euroopa-liidu-sonavoistlus-2014/> (30.11.2023)
- [3] IBM. What is the internet of things? <https://www.ibm.com/topics/internet-of-things> (01.12.2023)
- [4] M. Kahu, Lahendus avaandmete jagamiseks Cumulocity IoT platvormilt. Tartu Ülikool, Arvutiteaduste instituut, 2023. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=77589 (11.05.2024)
- [5] Cumulocity. Introduction to our IoT platform. <https://www.cumulocity.com/guides/concepts/introduction/> (29.11.2023)
- [6] Cybernetica AS. Andmekaitse ja Infoturbe Leksikon. Rakendusliides. <https://akit.cyber.ee/term/3088-rakendusliides-api>
- [7] Cumulocity. OpenAPI Specification. <https://cumulocity.com/api/core/10.18.0/> (11.05.2024)
- [8] InfluxData. InfluxDB Open Source. <https://www.influxdata.com/products/influxdb/> (29.11.2023)
- [9] Cybernetica AS. Andmekaitse ja Infoturbe Leksikon. Aegrida. <https://akit.cyber.ee/term/4986-aegrida>
- [10] Grafana Labs. About Grafana. <https://grafana.com/docs/grafana/latest/introduction/> (29.11.2023)
- [11] Grafana Labs. Visualizations. <https://grafana.com/docs/grafana/latest/panels-visualizations/visualizations/> (29.11.2023)
- [12] Grafana Labs. Dashboards. <https://grafana.com/docs/grafana/latest/dashboards/> (29.11.2023)
- [13] Grafana Labs. Alerting. <https://grafana.com/docs/grafana/latest/alerting/> (03.12.2023)
- [14] Grafana Labs. Explore. <https://grafana.com/docs/grafana/latest/explore/> (29.11.2023)

- [15] Unwin, A. Why Is Data Visualization Important? What Is Important in Data Visualization? *Harvard Data Science Review*, 2(1). 2020. <https://doi.org/10.1162/99608f92.8ae4d525> (30.11.2023)
- [16] Ülemiste keskus. Majaplaan. <https://www.ulemiste.ee/majaplaan/> (02.05.2024)
- [17] Computer Graphics and Virtual Reality Study Lab. Delta Building Visualization. <https://dbv-sc.weebly.com/> (02.12.2023)
- [18] A. Nikolajev, Delta Building Visualisation and Optimisation. Tartu Ülikool, Arvutiteaduste instituut, 2018. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=61825 (02.05.2024)
- [19] Computer Graphics and Virtual Reality Study Lab. DeltaVR. <https://gitlab.com/UT-CGVR/deltavr> (03.05.2024)
- [20] Delta Majaplaan. <https://deltakaart.weebly.com/> (03.05.2024)
- [21] Lin, S., Fortuna, J., Kulkarni, C., Stone, M., Heer, J. Selecting Semantically-Resonant Colors for Data Visualization. *Computer Graphics Forum*, 32(3). 2013. <https://doi.org/10.1111/cgf.12127> (30.11.2023)
- [22] Grafana Labs. Configure standard options. <https://grafana.com/docs/grafana/latest/panels-visualizations/configure-standard-options/> (03.11.2023)
- [23] Sánchez, M. N., Giancola, E., Soutullo, S., Gamarra, A. R., Olmedo, R., Ferrer, J. A., Jiménez, M. J. Suitability Evaluation of Different Measured Variables to Assess the Occupancy Patterns of a Building: Analysis of a Classroom of a School in Madrid during the COVID-19 Pandemic. *Energies*, 15(9). 2022. <https://doi.org/10.3390/en15093112> (15.11.2023)
- [24] Román-García, A. Z., Nieto, R., Hernandez, P. V., Rodriguez-Sanchez, M. C., Carrillo, M. G. Smart Building Digital Twin: Wireless Sensing and Actuation Architecture at Rey Juan Carlos University. 2023 *IEEE International Workshop on Metrology for Living Environment (MetroLivEnv)*. 2023. <https://doi.org/10.1109/MetroLivEnv56897.2023.10164028> (15.11.2023)
- [25] Cybernetica AS. Andmekaitse ja Infoturbe Leksikon. Rastergraafika. <https://akit.cyber.ee/term/11305-rastergraafika>

- [26] Cybernetica AS. Andmekaitse ja Infoturbe Leksikon. Vektorgraafika. <https://akit.cyber.ee/term/13441-vektorgraafika>
- [27] Mozilla. SVG: Scalable Vector Graphics. <https://developer.mozilla.org/en-US/docs/Web/SVG> (03.05.2024)
- [28] World Wide Web Consortium. The ‘path’ element. <https://svgwg.org/svg2-draft/paths.html#PathElement> (03.05.2024)
- [29] World Wide Web Technology Surveys. Usage statistics of JavaScript as client-side programming language on websites. <https://w3techs.com/technologies/details/cp-javascript> (03.05.2024)
- [30] Allen, J. G., MacNaughton, P., Satish, U., Santanam, S., Vallarino, J., Spengler, J. D. Associations of Cognitive Function Scores with Carbon Dioxide, Ventilation, and Volatile Organic Compound Exposures in Office Workers: A Controlled Exposure Study of Green and Conventional Office Environments. *Environmental health perspectives*, 124(6). 2016. <https://doi.org/10.1289/ehp.1510037> (03.05.2024)
- [31] Mozilla. HTTP response status codes. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status> (03.05.2024)
- [32] Google Lighthouse. Overview. <https://developer.chrome.com/docs/lighthouse/overview> (13.05.2024)
- [33] Let’s Encrypt. FAQ. <https://letsencrypt.org/docs/faq/> (13.05.2024)

Lisad

Lisa 1. Platvormi lähtekood

Platvormi komponentide lähtekoodid asuvad avalikult kättesaadavates GitHub-i repositooriumites. Veebirakenduse komponendi kood asub lingil <https://github.com/thugoo/client-delta-twin>. Andmete pärimiste teenuste ja API teenuse kood asub lingil <https://github.com/thugoo/api-delta-twin>.

Lisa 2. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Hugo Martin Teemus**,

1. Annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose
Reaalajas Delta keskuse õppe- ja teadushoone andmete visualiseerimine,

mille juhendaja on Pelle Jakovits,

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi
DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks
Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative
Commonsi litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost
reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja
kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega
isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Hugo Martin Teemus

15.05.2024