

TALLINNA POLÜTEHNIKUM

IT ja telekommunikatsiooni erialaosakond

Rebecca Laaneväli

**AUTODESK REVIT TARKVARAS JOONISTE
EKSPORTIMISE AUTOMATISEERIMINE
KASUTADES DYNAMOT**

Lõputöö

NOOREM TARKVARAARENDAJA

KTA-21E

Juhendaja: Kristo Kaasan

Tallinn 2023

AUTORIDEKLARATSIOON

Deklareerin, et käesolev lõputöö, mis on minu iseseisva töö tulemus, on esitatud Tallinna Polütehnikumi lõputunnistuse taotlemiseks noorem tarkvaraarendaja erialal. Lõputöö alusel ei ole varem eriala lõputunnistust taotletud.

Autor Rebecca Laaneväli (digiallkirjastatud)

Töö vastab kehtivatele nõuetele

Juhendaja Kristo Kaasan (digiallkirjastatud)

Sisukord

Sisukord	3
Lühendite ja mõistete loetelu	5
Sissejuhatus.....	6
1. Ettevõtte tutvustus.....	7
2. Teoreetiline osa.....	8
2.1. Valdkonna tutvustus.....	8
2.2. Ülesande püstitus.....	8
2.2.1. Süsteemi nõuete püstitus ja analüüs.....	9
2.2.2. Probleemi võimalikud lahendused.....	10
2.3. Valitud lahenduse tutvustus	11
3. Praktiline osa.....	12
3.1. Nõuetele vastava lahenduse projekteerimine	12
3.2. Koodi kirjutamine, programmeerimine.....	12
3.2.1. Andmepäring Autodesk Revit-ist failide nimetamiseks	12

3.2.2. Dwg failide eksportimine läbi Python koodi	15
3.3. Testimine.....	18
3.4. Kasutamine.....	19
4. Tulemused, järeldused, ettepanekud	20
Kokkuvõte.....	21
Kasutatud kirjandus	22

Lühendite ja mõistete loetelu

BIM – on ehitatavast hoonest infomudel või ehitusinformatsiooni modelleerimine (inglise keeles *building information model* või *building information modelling*) (Riigi Kinnisvara AS, 2023).

EHR – riiklik ehitisregister (Tarbijakaitse ja Tehnilise Järelevalve Amet, 2023).

Revit – on Autodeski poolt loodud BIM tarkvara, mis aitab arhitektidel, inseneridel ja konstruktoritel luua kõrgkvaliteetseid hooneid ja rajatisi (Autodesk, 2023).

Dynamo – on avatud lähtekoodiga liides Autodesk Revit-i tarkvara jaoks. Dynamo võimaldab arhitektidel luua algorütmide ja parameetride abil eripäraseid disaine ja automatsioone läbi visuaalse programmeerimise liidese (Wintour, 2023).

Python – on tõlgendatav, interaktiivne objektorienteeritud programmeerimiskeel (Python Software Foundation, 2023).

Dwg – on 2D või 3D joonise faililaiend, mis on koostatud Autodesk AutoCAD tarkvaraga (Sharpened Productions, 2023).

Sissejuhatus

Lõputöö eesmärk on automatiseerida jooniste nimetamine Revit-is kasutades Dynamo koodi nii, et see vastaks EHR tingimustele. Teema püstituse idee tuli sellest, et autoril on haridus ja kogemus ehituse alal ning on olnud vastutav kasutusloa taotlemisega ja teadlik ehitusprojekti jooniste üleslaadimise kitsas kohtadega.

EHR üleslaadimise keskkonnas on oluline, et faili nimed vastavad nende poolt seatud tingimustele ja automaat lugemise tõttu takistab kasvõi ühe faili ebakorrektnet nimetus kogu ehitusprojekti üles laadimist. Kuna ehituses on jooniste muutmine ehituse jooksul tavapärase, siis uuendatud jooniste failinimetus on sekundaarne mure arhitektile, kuid väga oluline hilisemale ehitusprojekti üles laadijale. Antud teema püstitus oli ka osa autori praktikast Usesoft AS-is.

Lõputöö raames lahendatakse jooniste nimetamine minimaalse kasutaja poolse info sisestamisega ja kasutades Revit tarkvaras olevat projektiinfot ja automaatselt tarkvara poolt antud väärtustega. Lõputöö kood koostatakse Revit-is olemasoleva Dynamo-s, mis on graafilise programmeerimise liides. Pärast jooniste nimetamise *script*-i jooksutamisega, kasutades Revit-isse sisse ehitatud Dynamo player-it, hõlbustab hilisemat pdf failide eksportimist.

1. Ettevõtte tutvustus

Usesoft AS on Eesti IT firma, mis keskendub äritarkvaralahendustele. Firma peamiseks klientideks on ehitus- ja tootmisvaldkonna ettevõtted, kes tegutsevad Eestis. Usesoft on olnud tegevuses juba üle 30 aasta, mil see asutati 1991. aastal. Firmas on tööl enam kui 20 spetsialisti ja konsultanti. (Usesoft AS, 2023)

Ehitusvaldkonna tarkvara lahendustest pakub Usesoft peamiselt Autodesk-i projekteerimistarkvara, olles Autodesk-i Autoriseeritud edasimüüja, koolituskeskus ja ametlik sertifitseerimiskeskus. Autodesk pakub palju erinevaid tarkvarasid, millest Usesoft müüb enamusest ja osasid ka koolitab. (Usesoft AS, 2023)

Autor töötas praktika ja lõputöö ajal peamiselt Autodesk Revit-is oleva Dynamo lahendustega, mis automatiseeris programmi siseseid käske.

2. Teoreetiline osa

2.1. Valdkonna tutvustus

Ehitusel kasutatav tarkvara on peamiselt seotud projekteerimise eesmärgiga. Viimastel aastatel on saanud üha olulisemaks mudelprojekteerimine, mis aitab kaasa BIM arengule Eestis. Kuna mudelprojekteerimine on mahukas töö, siis korduvprotsesside automatiseerimine aitab kiirendada tööd ja vähendada inimeksimusi.

2.2. Ülesande püstitus

Lõputöö teema valik oli seotud autori eelneva hariduse ja töökogemusega ehitussektoris, ning praktikal püstitatud ülesandega. Kuna autor on kokku puutunud eelneval tööol ehitusdokumentatsiooni esitamisega EHR keskkonnas kasutusloa saamise eesmärgil. Selle tegevuse käigus tuli ilmsiks, kuidas on probleeme ehitusjooniste nimetamisega nii, et keskkond lubaks neid üles laadida. Kasutusloa taotlemisel on keerulisim hetk, kui on vaja tööprojekt esitada EHR-i e-keskkonnas. Ehitusprojekti dokumendi liigi puhul on keskkonnas sisse ehitatud automaatkontroll, mis kontrollib kõikide failide nimetused üle. Seetõttu oli autoril hea ülevaade probleemist ning praktika koha eripärast tulenevalt, sobis selle lahendamine üheks osaks praktikal.

Antud probleemile lahenduse leidmine kiirendaks arhitektide ja konstruktorite tööd, kuna ehituse käigus parandatakse jooniseid jooksvalt kuni hoone valmimiseni. Korrigeeritud

jooniste esitamine koos õigete failide nimetustega, mis läheb hõlpsalt EHR keskkonda üles, elimineerib arhitekti hilisemist parandamist käsitsi.

Lahendus võimaldab muuta jooniseid, muudatuste käigus määrata uus nimetus, mis vastaks EHR-i nõuetele, ja eksportida dwg failid eelnevalt mainitud nimetusega.

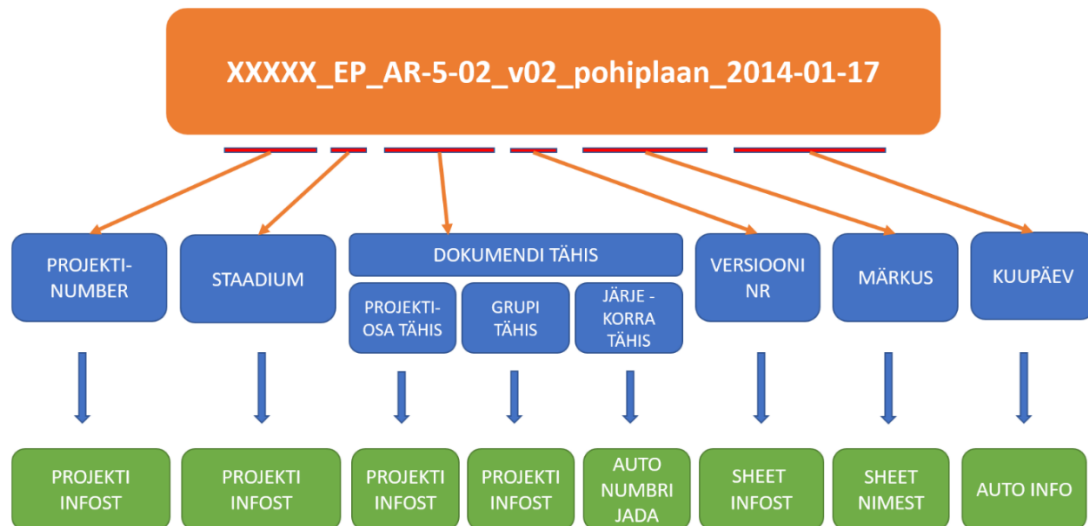
2.2.1. Süsteemi nõuete püstitus ja analüüs

EHR keskkonnas on kindel ülesehitus, millest faili nimetus peab koosnema. EHR on andnud välja juhendi „Ehitusprojekti failide vormistamise juhend“ ning nende nõuete kaudu selgitas autor välja, mis info ehitusjooniste nimetuse puhul vajalik on.

Ehitusprojekti üles laadimisel kasutatakse konteiner viisi, ehk vajalikud projektiosad, mis on allkirjastatud, pakendatakse kokku. Ehitusprojekti joonise faili nimetus koosneb (Ehitisregister, 2023):

- projektinumbrist;
- projektistaadiumist;
- projektiosa tähisest;
- joonise grupi tähisest;
- joonise järjekorra tähisest;
- versiooni numbrist;
- märkusest (näiteks „1. korruse põhiplaan“);
- kuupäevast.

Kuna ülesande eesmärk oli kiirendada ehitusjooniste väljastamist ning et ka faili nimetused vastaks EHR-i nõuetele, siis lahendati probleem läbi Autodesk Revit-i. Autodesk Revit võimaldab läbi programmi kasutada Dynamo-t, mida saab kasutada käskude automatiseerimiseks. Seetõttu on oluline analüüsida, kuidas saada Revit-i kaudu vajalik info, et koostada õiged failide nimetused. Autor koostas selle jaoks põhimõttelise joonise, kust saada kätte vajalikud andmed (**Joonis 1**).



Joonis 1. Programmist Revit andmete kogumine faili nimetamiseks.

2.2.2. Probleemi võimalikud lahendused

Kuna ülesande eesmärk on lihtsustada arhitektide jaoks failide nimetamist hilisemas projekti staadiumis, siis tulenevalt Autodesk Revit-i populaarsusest Eestis ja praktikakohast, oli vajalik ülesanne lahendada Autodesk-i siseses programmis. Revit-is on võimalik kasutada Dynamo tarkvara, mis on peamiselt visuaalsele programmeerimisele üles ehitatud.

Dynamo-s on võimalik andmeid saada mitut viisi. Ülesande eesmärk oli kasutada vaid *node*, mis on juba vaikimisi allalaetud Dynamos.

Dynamo *node*-idel on kolm põhilist käitumist (Rezk, 2023):

- loomine – loo andmed või geomeetria;
- modifitseerimine – kalkuleeri andmetega;
- andmete päring – saa informatsiooni.

Kuna Dynamo on visuaalne programmeerimine, siis sai jooniste nimetamise osa tehtud visuaalsete käsklustega. Jooniste failide lõplik formaat koosneb mitmest osast, mis on vaja

läbi Revit programmi andmetest saada. Seetõttu visuaalselt andmete päringu ja modifitseerimist õigeks kujuks on iga info osa kaupa näidatud *node*-ide abil info asukohta andmesüsteemis.

Pärast joonistele õige faili nimetuse andmist, on vaja eksportida välja joonised dwg formaadis. Läbi Dynamo on selline automaatne käsklus puudu ilma, et oleks vaja kasutada kolmandate osapoolte poolt koostatud pakette. Selle ülesande osa lahendamine oli ainult võimalik koostades Python-i koodi, mis teostaks vajalikud funktsioonid, et iga dwg fail oleks nimetatud vastavalt uuele joonise faili nimetusele, mis vastaks EHR-i tingimustele.

2.3. Valitud lahenduse tutvustus

Ülesanne lahendatakse läbi Autodesk Revit-is oleva liidese Dynamo kaudu. Failide nimetamise osa lahendamisel kasutatakse Dynamo-s olevaid *node*. Ülesande käigus ei kasutata kolmandate osapoolte poolt tehtud pakette, et lahendus oleks kasutajasõbralikum. Kui nimetamisel on võimalik kõik info saada läbi Dynamo *node*-ide, siis dwg failide eksportimine koos valitud faili nimega, on võimalik vaid Python-i koodi koostamisega. Python-i kasutamine ülesande teises osas elimineerib vajadust kasutada kolmandate osapoolte pakette.

3. Praktiline osa

3.1. Nõuetele vastava lahenduse projekteerimine

Ülesande esimene etapp hõlmas Autodesk Revit-i kaudu projekti info saamist. Revit-isse sisse ehitatud Dynamo abil saab teha andmehulgale päringu ja selle info kohaselt koostada nimekiri jooniste nimetustest, mis vastaks failide üles laadimisel EHR-i tingimustele.

Pärast edukat andmepäringut faili nimetuse saamist toimub ülesande teine etapp. Järgmises etapis kasutatakse eelnevalt koostatud nimekirja jooniste eksportimisel dwg formaati. Ülesanne oli edukalt lahendatud kui eksportimisel oli õige arv jooniseid dwg formaadis ja nende nimetused vastasid EHR nõuetele.

3.2. Koodi kirjutamine, programmeerimine

3.2.1. Andmepäring Autodesk Revit-ist failide nimetamiseks

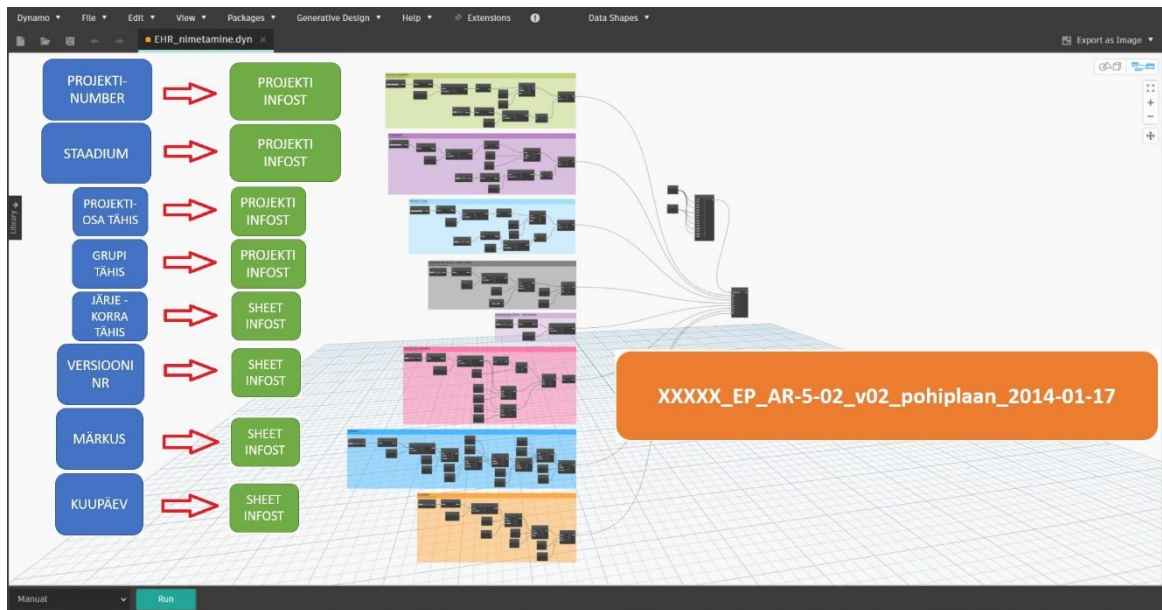
Enne koodi kirjutamist oli vajalik täita ehitusprojekti kohane info. Projekti kohta info sisestamine on tavapärane tegevus arhitektidel ja konstruktoritel, ning seetõttu pole sisestus lisasamm kasutajale.

Kuigi projekti kohta on võimalik Autodesk Revitis sisestada mitmeid parameetreid, siis edukaks andmepäringu ja nimetamise protsessiks oli vajalik täita(**Joonis 1**):

- projektinumber;
- staadium;
- projektiosatähis;
- grupitähis.

Ülesande lahendamise keskmes olid 2D joonised, ning seetõttu oli vajalik Autodesk Revit keskkonnas vormistada joonised. Jooniste vormistamisel on välja antud standardid, mille järgi koostatakse kirjanurk joonisel (Maikov, 2023).

Failide nimetamisel oli vaja saada jooniste (Sheet) infost järjekorratähis, versiooni number, joonise nimetust ehk märkus ja kuupäev. Autor koostas visuaalse lahenduskäigu, näitamaks iga andmekogu päringut(**Joonis 2**). Järjekorratähise annab Revit tarkvara automaatselt, kui paberkeskkond joonise jaoks luuakse. Versiooni number kajastab mitmes versioon on joonisest väljastatud ning selle info salvestab Revit iga joonise kohta eraldi *Revisions on Sheet* parameetri alla. Joonise nimetust kasutatakse ka kirjanurgas, millel on kindel parameeter nimega *Sheet Name* ja selle parameetri abil saame faili nimetamiseks märkuse iga joonise kohta. Joonise kuupäev salvestatakse joonise info hulgas ja näitab, mis kuupäeval on joonis loodud. Dwg faili nimetuse saamiseks kasutatakse Revit-i liidendit Dynamo.



Joonis 2. Dynamo-s failide nimetamise osade ülesehitus.

Kuna nimetus koosneb erinevatest infopäringutest Revit keskkonnast, siis on andmete kogumine jagatud Dynamo-s kaheksaks osaks. Iga saadud infokogum oli vajalik viia vastavasse formaati, et joonise failinimetus oleks korrektne EHR nõuetele. Iga osa lahendamiseks kasutati Dynamo-s olevaid *node*.

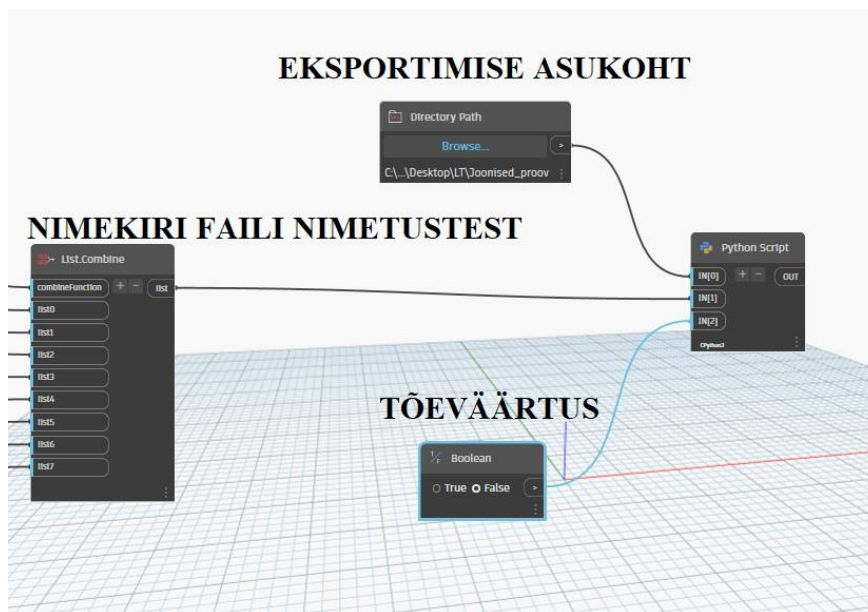
Infokogumid, mida oli vaja viia õigesse formaati, olid jooniste grupitähis, joonise märkus ehk joonise nimetus ja kuupäev. Joonise grupitähis annab infot, kas joonis on horisontaalne ehk siis grupitähises 5 või vertikaalne ehk grupitähises 6. Selle lahendamiseks kasutasin joonise kirjanurgast olevat nimetust, kuna horisontaalsetel arhitektuursetel joonistel kasutatakse sõna „plaan“. Eelnevalt mainitud sõna kaudu jagasin joonised kaheks - joonised, millel on antud sõna joonise nimetuses ja joonised, millel sõna pole. Järgmisena oli vaja korrigeerida joonise kirjanurgas olevat märkust. Kuna eesti keeles on erinevad täpitähed, siis vahetati need sümbolite vastu, mis olid tähtedele alternatiivid ladina tähestikust. Samuti asendati märkuses ühikud sidekriipsuga. Enne infode kombineerimist oli muudeti kuupäeva formaat nii, et esimesel kohal oleks aasta, teisel kohal kuu ja viimasel päev, ning need oleksid omavahel seotud sidekriipsuga.

Iga infokogu eraldati alakriipsuga, välja arvatud projektiosa, grupi ja järjekorra tähis, mis eraldati sidekriipsuga. Faili nimetuse koostamiseks kasutati nimekirja kombineerimise *node*-i, mis võimaldas täpsustada, kuidas uus nimekiri iga elemendiga koostatakse. Kombineerimise funktsiooniks kasutati *string* aheldamise *node*-i nii, et iga osa eraldaja sai määratud oma asukoha põhiselt, jättes tühjaks nimetajad, kuhu kombineerimisel lisatakse projektis olev info iga joonise kohta. Kogu andmepäringu jooksumisel Dynamo-s, tekib nimekiri jooniste dwg failide nimetustest.

3.2.2. Dwg failide eksportimine läbi Python koodi

Kuna Autodesk Revit keskkonnas ei ole võimalik dwg faile eksportida nii, et nimetamisel kasutatakse uut reeglistiku, siis koostati Dynamo siseselt uus nimekiri jooniste failide nimedest. Kui ülesande esimeses osas oli võimalik vajalikud tegevused teostada Dynamo-s olevate *node*-idega, siis dwg eksportimise funktsiooni pole välja ehitatud nii, et ei peaks alla laadima kolmandate osapoolte väljastatud pakette. Seetõttu lahendati dwg eksportimise võimalus läbi Python *node*-i Dynamo-s.

Python-it kasutades tekkisid probleemid peamiselt sellega, et Autodesk Revit ei ole väljastanud piisavat dokumentatsiooni Python-i kasutamiseks, mis oleks lihtsustanud võimalikkude funktsioonide avastamist. Peale eksportimise, tegi Python ka andmepäringu Revit tarkvarasse seoses projektis koostatud jooniste paberkeskkondade kohta. Seetõttu oli vajalik aru saada, kus kategoorias ja klassis otsitav element asub. Pythoni koodi toodi andmeid ka läbi Dynamo-s olevate sisendite kaudu. Läbi sisendite toodi sisse - asukoht, kuhu failid salvestatakse, failide nimetamiseks eelnevalt loodud nimekiri ja tõeväärtus selleks, et jooniste eksportimisel tuuakse välja vaid uue nimekirja järgi, mitte Revit-is vaikimisi olevate reeglite järgi.



Joonis 3. Sisendandmed Python koodi Dynamo *node*-idega.

Enne koodi kirjutamist on vaja Python-isse sisse tuua vajalikud moodulid (**Joonis 4**), mis võimaldavad kasutada lahendamiseks vajalikke käsklusi. Esimene moodul, mis sisse oli vaja tuua, oli „clr“. Põhjusega, et saaksime Revit tarkvaras olevaid nimeruume kasutada nii nagu tavalisi Python moodulites (Autodesk, 2023). Sisse toodud mooduli abil saab koodis kasutada Revit süsteemis olevaid funktsioone (Autodesk, 2023). „Clr“ mooduli kaudu anname koodile ligipääsu Revit tarkvaras olevatele andmekogudele ja käsklustele, mis võimaldab muuta projektis olevat infot. Oluline moodul, mis võimaldab kasutada Python-i süsteemi, on „sys“ moodul (Python Software Foundation, 2023). Selle mooduli kaudu täpsustati, kus asub Python-i andmekogud variatsioonidest ja funktsioonidest. Viimase moodulina on sisse toodud „subprocess“, mille kaudu kasutati funktsiooni avamaks kausta, mis anti koodis sisendiks, pärast koodi jooksutamist (Python Software Foundation, 2023).


```

4 import clr
5 import sys
6 import subprocess
7
8 clr.AddReference("RevitAPI")
9 clr.AddReference("RevitServices")
10
11 from Autodesk.Revit.DB import *
12 from RevitServices.Persistence import DocumentManager
13 from System.Collections.Generic import List
14
15 path = r'C:\Python310\Lib'
16 sys.path.append(path)
17

```

Joonis 4. Kasutatud moodulid Python koodis.

Pärast moodulite importimist, täpsustati koodis variatsiooni väärtusi (**Joonis 5**). Failide nimed toodi sisse sisend[1] kaudu ja seda tähistati koodis *names* variatsioonina. Sisendväärtusena[0] täpsustati asukohta seadmes, mille väärtus hoiustati *folderPath*-is. Koodi eesmärk oli kasutada infot konkreetses projektis, mis on Revit tarkvaras avatud. Seetõttu täpsustati koodis, et andmepäringul otsitakse ainult hetkel lahti olevas projektis. Kui *sheet* variatsioonis kasutatakse filtreerimist, siis täpsustati andmekogu, milles protsess toimub (Autodesk, 2023). Antud variatsiooni kogutakse paberkeskkonnas koostatud jooniseid, ehk limiteeritakse andmepäring klassis *ViewSheet*. Saadud tulemus filtreeritakse *ToElements* meetodi kaudu, mis tagastab kogumiku elementidest (Autodesk, 2023). Viimasena tehakse tühi nimekiri nimega *result*.

```

18 doc = DocumentManager.Instance.CurrentDBDocument
19
20 folderPath = IN[0]
21
22 names=IN[1]
23
24 sheets=FilteredElementCollector(doc).OfClass(ViewSheet).ToElements()
25
26 result=[]
27

```

Joonis 5. Variatsioonid Python koodis.

Pärast failinimetuste ja paberkeskkonna jooniste kogumist, on vajalik igale joonisele anda vastav nimetus. Kuna kood kasutab samas dokumendis olevaid elemente selleks, et

eelnevalt mainitud andmepäringuid teha, siis võib eeldada, et mõlemas variatsioonis, nii *sheets* kui ka *names*, nimekirjas oli sama palju indekseid. Seetõttu kasutati koodis Python *zip()* funktsiooni, et nimekirjasid paralleelselt läbi töötada (Ramos, 2023) (**Joonis 6**). Kuna sooviti kasutada funktsiooni *Export()*, oli vajalik anda funktsioonile ette – *string* kaust, *string* nimi, *ElementId*, *DWGExportOptions* (Autodesk, 2023). Sisendina tulnud kausta väärtus oli juba *string* vormis. Samuti ei vaja korrigeerimist variatsioon *n*, kus olid failide nimetuste andmed. Kuna *Export()* funktsioon vajab *Sheet* elemendi id väärtust, siis oli vajalik enne teha uus nimekiri *sheetsE*, mis tegi päringu andmebaasi ja väljastas jooniste id väärtused. Klassile *DwgExportOptions* on antud eraldi variatsioon selleks, et muuta *MergedViews* vaikimis olevat väärtust (Autodesk, 2023). Eelnevalt mainitud väärtus muudeti sisend tõeväärtusega õigeks selleks, et Revit ekspordiks vaid variatsioonis *n* nimetustega dwg faile. Pärast ekspordimist lisatakse väärtused *result* nimekirja. Viimase funktsioonina avas kood kausta, kuhu antud failid eksporditi.

```
28 for sheet, n in zip(sheets, names):
29     options=DWGExportOptions()
30     options.MergedViews=IN[2]
31     sheetsE = List[ElementId]()
32     sheetsE.Add(sheet.Id)
33     result.append(doc.Export(folderPath, n, sheetsE, options))
34
35 subprocess.Popen(['explorer', IN[0]])
36
37 OUT=result
```

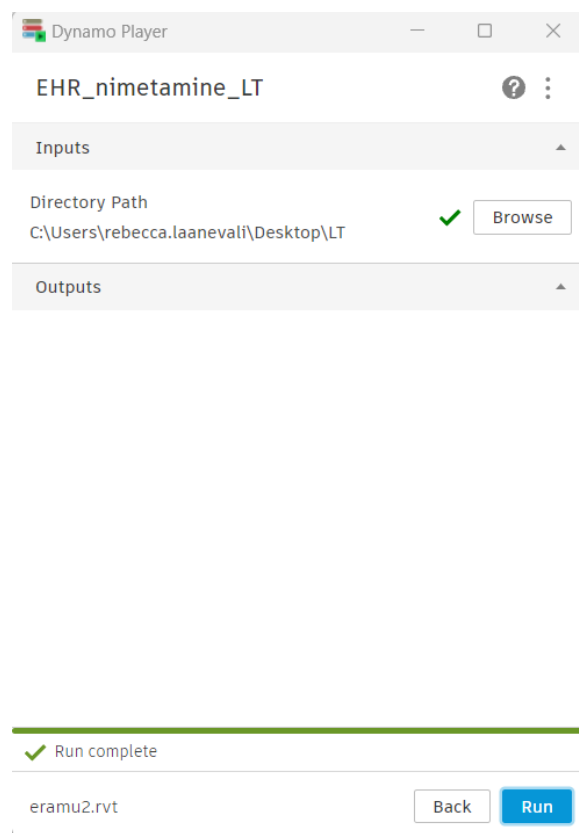
Joonis 6. Failide eksport Python koodis.

3.3. Testimine

Koodi kirjutamise käigus kasutati Dynamo-s olevat *node*-i nimega *Watch*. Mainitud võimalust kasutati kui teostati andmepäringuid. Kasutades seda funktsiooni sai kontrollida, mis info Revit keskkonnast üle tuleb ja mis kujul. Samuti oli võimalik igast *node*-ist näha, mis andmed olid läbinud. Python-i *script*-i testimiseks kasutati Dynamo-s olevat koodi läbi jooksutamist. Kui kood ei läbinud edukalt, väljastas Python *node* error sõnumi. Kuna antud koodi eesmärk oli seda kasutada Revit keskkonnas, siis oli vajalik testida, kas antud lahendus töötab Dynamo player-i, mis võimaldab koodi jooksutada Revit keskkonnas. Edukas väljund koodil oli dwg failide loomine koos nimetustega, mis läbi andmepäringu koostati. Antud lahendus võimaldas saavutada soovitud väljundit.

3.4. Kasutamine

Antud lõputöö raames koostatud lahendust on mõeldud kasutamiseks Revit tarkvara keskkonnas. Tarkvaras on sisse ehitatud Dynamo player, mis võimaldab kasutada koodi, mis on teostatud Dynamo-s (**Joonis 7**). Koodi kasutamiseks on vajalik omada koodi faili. Dynamo player-is on vajalik täpsustada asukoht, kus kood asub, ja seejärel kuvab liides kaustas olevad Dynamo-s koostatud faile. Valides autori koostatud koodi, on kasutajal vajalik vaid täpsustada tulevaste failide asukoht. Koodi jooksumisel, avatakse ka pärast edukat läbimist kaust, mille kasutaja määras.



Joonis 7. Dynamo player Revit tarkvaras.

4. Tulemused, järeldused, ettepanekud

Lahenduse loomise eesmärk oli automatiseerida dwg failide eksportimist Revit tarkvaras nii, et failide nimetused vastaksid EHR nõuetele projekti üles laadimisel. Lõputöö käigus loodud kood võimaldab neid eesmärke täita. Antud kood ekspordib kõik Revit paberkeskkonnas loodud joonised.

Kuigi lahendus toimib, oleks kasumlik arendada edasi võimalust eksportida vaid üksikuid jooniseid. Samuti, kuna eksportimise käsklus võtab aega ning reaalses olukorras on failide hulk suurem, tuleks tulevikus edasi arendamisel pöörata ka tähelepanu sellele, kas on võimalik protsessi kiirendada.

Kokkuvõte

Lõputöö käigus sai lahendatud jooniste nimetamise automatiseerimine Revit tarkvaras selleks, et failide hilisem üleslaadimine EHR keskkonda oleks probleemi vaba. Lahendus sai koostatud Revit-i tarkvaras olemasoleva Dynamo graafilises programmeerimise liideses. Lahenduse koostamisel kasutati vaid Dynamo-s vaikimisi olemasolevaid `node`-e selleks, et kood oleks kasutajasõbralikum ning ei eeldaks kolmandate osapoolte pakette.

EHR-i nimetamise tingimused said jaotatud 8 osaks, mille väärtused saadi Revit-is olevast projektiinfost või mis tarkvara ise automaatselt andis. Jooniste failide nimetuste koostamisel kasutati Dynamo-s vaikimisi olemasolevaid `node`. Eksportimise funktsioon lahendati läbi Python `node`-i.

Lahenduse kasutamiseks vajab kood kasutajalt vaid tulevaste failide eksportimise asukohta. Lõputöö eesmärgil loodud koodi jooksutamiseks kasutati Revit tarkvaras olevat Dynamo player-it. Pärast koodi rakendamist olid jooniste nimetus vastav EHR tingimustele.

Kasutatud kirjandus

Autodesk. (28. 04. 2023. a.). DWGExportOptions Properties. Allikas:
<https://www.revitapidocs.com/2023/b0f9d024-798f-e97d-896e-379b324d4dd5.htm>

Autodesk. (23. 04. 2023. a.). Export Method. Allikas:
<https://www.revitapidocs.com/2023/44ee91ff-c9f3-7df5-b8c0-81c17ac75dc7.htm>

Autodesk. (28. 04. 2023. a.). FilteredElementCollector class. Allikas:
<https://www.revitapidocs.com/2023/263cf06b-98be-6f91-c4da-fb47d01688f3.htm>

Autodesk. (24. 04. 2023. a.). Python and Revit. Allikas:
https://primer.dynamobim.org/10_Custom-Nodes/10-5_Python-Revit.html

Autodesk. (03. 04. 2023. a.). Revit. Allikas:
<https://www.autodesk.eu/products/revit/overview?term=1-YEAR&tab=subscription>

Autodesk. (28. 04. 2023. a.). ToElements Method. Allikas:
<https://www.revitapidocs.com/2023/732b4a0d-62d8-b86d-120b-8ea3d9713b34.htm>

Ehitisregister. (03. 04. 2023. a.). *Rakenduste kasutusjuhendid*. Allikas: EHR:
https://livekluster.ehr.ee/ui/ehr/v1/help/instructions/Ehitusprojekti_failide_vormistamise_juhend.pdf

Maikov, K. (24. 04. 2023. a.). Jooniste vormistamine. Tartu, Tartumaa, Eesti. Allikas:
<https://sites.google.com/site/tehnilinejoonestamine/jooniste-vormistamine>

Python Software Foundation. (03. 04. 2023. a.). General Python FAQ. Allikas:
<https://docs.python.org/3/faq/general.html#what-is-python>

Python Software Foundation. (24. 04. 2023. a.). subprocess - Subprocess management. Allikas: <https://docs.python.org/3/library/subprocess.html>

Python Software Foundation. (24. 04. 2023. a.). sys — System-specific parameters and functions. Allikas: <https://docs.python.org/3/library/sys.html>

Ramos, L. P. (28. 04. 2023. a.). Using the Python zip() Function for Parallel Iteration. Allikas: <https://realpython.com/python-zip-function/>

Rezk, S. (03. 04. 2023. a.). Getting Started with Dynamo for Civil 3D: A Beginner's Guide. Allikas: <https://www.autodesk.com/autodesk-university/article/Getting-Started-Dynamo-Civil-3D-Beginners-Guide-2022>

Riigi Kinnisvara AS. (03. 04. 2023. a.). Panus ühiskonda BIM. Allikas: <https://www.rkas.ee/et/panus-uhiskonda/bim>

Sharpened Productions. (28. 04. 2023. a.). .DWG File Extension. Allikas: <https://fileinfo.com/extension/dwg>

Tarbijakaitse ja Tehnilise Järelevalve Amet. (03. 04. 2023. a.). Riiklik ehitisregister (EHR). Allikas: <https://ttja.ee/ariklient/ehitised-ehitamine/ehitisregister-ehr>

Usesoft AS. (03. 04. 2023. a.). Ettevõttest. Allikas: <https://usesoft.ee/ettevotest/>

Wintour, P. (03. 04. 2023. a.). What is Dynamo? Allikas: <https://parametricmonkey.com/2015/02/21/what-is-dynamo/>

