Ref. Ares(2021)4241912 - 29/06/2021

BIMERR
RENOVATION 4.0

Project Acronym:	**BIMERR**
Project Full Title:	**BIM-based holistic tools for Energy-driven Renovation of existing Residences**
Grant Agreement:	**820621**
Project Duration:	**45 months**

DELIVERABLE D7.8
Urban Planning Module 2

Deliverable Status:	**Final**
File Name:	**BIMERR_D7.8-v1.00**
Due Date:	**30/06/2021 (M30)**
Submission Date:	**30/06/2021 (M30)**
Task Leader:	**Xylem (T7.4)**

Dissemination level	
Public	x
~~Confidential, only for members of the Consortium (including the Commission Services)~~	

This project has received funding from the European Union's Horizon 2020 Research and innovation programme under Grant Agreement n°820621

The BIMERR project consortium is composed of:		
FIT	Fraunhofer Gesellschaft Zur Foerderung Der Angewandten Forschung E.V.	Germany
CERTH	Ethniko Kentro Erevnas Kai Technologikis Anaptyxis	Greece
UPM	Universidad Politecnica De Madrid	Spain
UBITECH	Ubitech Limited	Cyprus
SUITE5	Suite5 Data Intelligence Solutions Limited	Cyprus
HYPERTECH	Hypertech (Chaipertek) Anonymos Viomichaniki Emporiki Etaireia Pliroforikis Kai Neon Technologion	Greece
MERIT	Merit Consulting House Sprl	Belgium
XYLEM	Xylem Science and Technology Management GmbH	Austria
CONKAT	Anonymos Etaireia Kataskevon Technikon Ergon, Emporikon Viomichanikonkai Nautiliakon Epicheiriseon Kon'kat	Greece
BOC	Boc Asset Management GmbH	Austria
BX	Budimex S.A.	Poland
UOP	University Of Peloponnese	Greece
UEDIN	University Of Edinburgh	United Kingdom
UCL	University College London	United Kingdom
NT	Novitech As	Slovakia
FER	Ferrovial Agroman S.A.	Spain

Disclaimer

BIMERR project has received funding from the European Union's Horizon 2020 Research and innovation programme under Grant Agreement n°820621. The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Commission (EC). EC is not liable for any use that may be made of the information contained therein.

Authors List

Leading Author (Editor)				
Surname		First Name	Beneficiary	Contact email
Bergmayr		Julia	Xylem	bergmayr@xylem.tech
Co-authors (in alphabetic order)				
#	**Surname**	**First Name**	**Beneficiary**	**Contact email**
1	Fenz	Stefan	Xylem	fenz@xylem.tech
2	Giannakis	Giorgos	Hypertech	g.giannakis@hypertech.gr
3	Heurix	Johannes	Xylem	heurix@xylem.tech
4	Manesis	Fotis	CONKAT	fmanesis@konkat.gr
5	Neubauer	Thomas	Xylem	neubauer@xylem.tech
6	Papapolyzos	Thomas	Hypertech	thomas@hypertech.gr
7	Wachter	Christoph	Xylem	wachter@xylem.tech
8	Wellner	Florian	Xylem	wellner@xylem.tech

Reviewers List

List of Reviewers (in alphabetic order)				
#	Surname	First Name	Beneficiary	Contact email
1	Lucerski	Maciej	BX	maciej.lucerski@budimex.pl
2	Tsoulos	George	UOP	gtsoulos@uop.gr

Revision Control

Version	Author	Date	Status
0.01	Xylem	17.11.2020	Draft ToC and Introduction
0.10	Xylem, CONKAT and Hypertech	11.01.2021	Draft Chapter 2, 3, and 4
0.20	Xylem, CONKAT and Hypertech	11.05.2021	Chapter 2
0.30	Xylem	31.05.2021	Chapter 3 and 4
0.40	Xylem	07.06.2021	Quality Check
0.90	Xylem	25.06.2021	Revision after internal review
1.00	Xylem	30.06.2021	Submission to the EC

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

ACRONYMS

Acronym	Meaning
BEP	Building Energy Performance
BIF	BIMERR Interoperability Framework
BIMERR	BIM-based holistic tools for Energy-driven Renovation of existing Residences
GML	Geography Markup Language
IFC	Industry Foundation Classes
KPI	Key Performance Indicator
RenoDSS	BIMERR Renovation Decision Support System

EXECUTIVE SUMMARY

This document describes the BIMERR Deliverable D7.8 "Urban Planning Module 2" demonstrator and concludes the final iteration of the development activities in T7.4 "Urban Planning module". This final release of the BIMERR Urban Planning module provides the geographical perspective of the building under renovation and its surrounding buildings, detailed information about energy production/consumption patterns and how the building interacts with utility networks in its vicinity.

In the final release of the BIMERR Urban Planning module we (i) integrated energy network data and visualized energy networks based on sample data, (ii) implemented CityGML export download functionality to ensure that available building and energy network data can be downloaded by the user in a standardized data format, (iii) visualized excess energy on the district level, (iv) implemented an automated weather data retrieval to support the building energy performance calculations with accurate weather data, and (v) visualized the current building and renovated scenarios based on the corresponding IFC files (based on third-party libraries).

The BIMERR Urban Planning module is based on state-of-the-art technologies and three layers: (i) the Presentation Layer which visualizes the energy flows and allows users to set their preferred data sources for energy production/consumption profiles of surrounding buildings. The user interface is built upon Angular, Typescript, and NGRX Entity/Store and uses Leaflet for visualizing the map view, (ii) the Business Logic Layer which calculates the KPIs and energy flows based on the stored production/consumption profiles of the renovation project and the surrounding buildings. This layer is based on Spring Boot and OpenJDK, and (iii) the Data Layer utilizes PostgreSQL to store consumption/production profile data of the renovation project and surrounding buildings for faster access at later usage.

1. INTRODUCTION

The BIMERR Urban Planning module provides the geographical perspective of the building under renovation (renovation project) and its surrounding buildings based on the geolocation extracted from the building's IFC[1] file. In the final release of the BIMERR Urban Planning module we (i) integrated energy network data and visualized energy networks based on sample data, (ii) implemented CityGML export download functionality to ensure that available building and energy network data can be downloaded by the user in a standardized data format, (iii) visualized excess energy on the district level, (iv) implemented an automated weather data retrieval to support the building energy performance calculations with accurate weather data, and (v) visualized the current building and renovated scenarios based on the corresponding IFC files (based on third-party libraries).

1.1 Scope and Objectives of the Deliverable

D7.8 "Urban Planning Module 2" reports the development activities in the context of Task T7.4 "Urban Planning module" of WP7 "Renovation Decision Support System". It documents the final version of the BIMERR Urban Planning module.

The objective of this document is to give an overview and documentation of the final stable release of the BIMERR Urban Planning module and describe:

- Functionalities of the BIMERR Urban Planning module
- Technology stack
- Communication with the BIMERR data management module
- Assumptions and restrictions of the final release
- Installation instructions
- Usage walkthrough

[1] https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/, last access: 24.08.2020

- Licensing

1.2 Relation to other tasks/deliverables

T7.4 "Urban Planning module" and therefore D7.8 "Urban Planning Module 2" are related to the following BIMERR deliverables:

- D3.1 "Stakeholder requirements for the BIMERR system": the business scenarios, use cases, and system requirements described in D3.1 are the basis for the development of the BIMERR Urban Planning module.
- D3.3 "BIMERR evaluation methodology": energy KPIs related to Urban Planning described in D3.3 are calculated by the Urban Planning module.
- D3.6 "BIMERR system architecture final version": the final version of the BIMERR architecture provided an overview on the BIMERR components, how they communicate to each other and how the BIMERR Urban Planning module is embedded in the BIMERR Renovation Support Tools.
- Functionalities that are introduced in this deliverable will be integrated into the Renovation Decision Support System (RenoDSS) developed in T7.5.
- T8.1 "External Information Availability and Sourcing": ensures that energy network and surrounding building data relevant for pre-validation and validation sites are available.
- D9.1 "Pilot renovation sites acquisition/selection process, ex-ante analysis and baseline definition"

2. BIMERR Urban Planning module

2.1 Overview

The BIMERR Urban Planning module provides the geographical perspective of the building under renovation and its surrounding buildings, detailed information about energy production/consumption patterns and how the building interacts with utility networks in its vicinity.

The BIMERR Urban Planning module is embedded in BIMERR RenoDSS and available at: https://renodss.xylem-technologies.com/ [2]

2.2 Architecture

Based on the BIMERR architecture (see Deliverable D3.6), Figure 1 shows the architecture of RenoDSS and its modules. RenoDSS communicates via the RenoDSS Data Management Module with the BIMERR Urban Planning module. The Data Management Module sends the IFC file and the BEP module's output to the Urban Planning module, which returns the district energy flow KPIs, visualizes the building and its surrounding on a map, provides tools to enter energy production/consumption patterns of surrounding buildings and provides a CityGML export containing relevant building and energy network data.

[2] Please send an email to support@xylem-technologies.com to request access credentials.

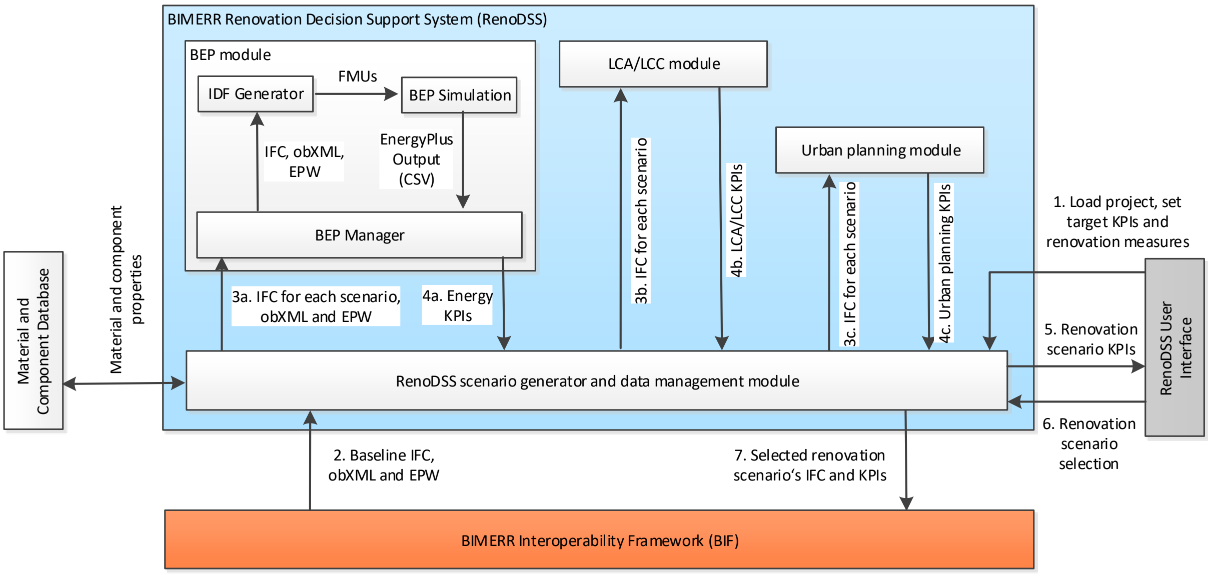

Figure 1: Architecture of the BIMERR Renovation Support Tools

2.3 Technology Stack and Implementation Tools

The BIMERR Urban Planning module is based on state-of-the-art technologies and three layers (see Figure 2):

- The Presentation Layer which visualizes the energy flows and allows users to set their preferred data sources for energy production/consumption profiles of surrounding buildings. The user interface is built upon Angular, Typescript, and NGRX Entity/Store and uses Leaflet for visualizing the map view.
- The Business Logic Layer which calculates the KPIs and energy flows based on the stored production/consumption profiles of the renovation project and the surrounding buildings. This layer is based on Spring Boot and OpenJDK.
- The Data Layer utilizes PostgreSQL to store consumption/production profile data of the renovation project and surrounding buildings for faster access at later usage.

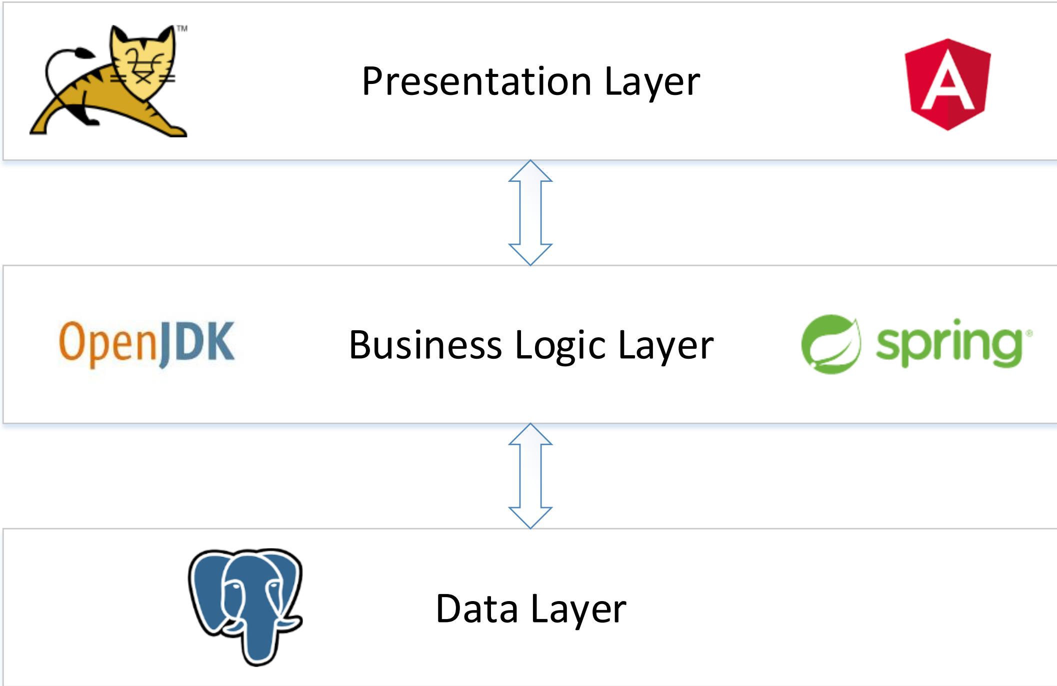

Figure 2: Architecture of the BIMERR Urban Planning module

In more detail, the BIMERR Urban Planning module utilizes open-source technologies and libraries as defined in the following table.

Table 1: Technologies and libraries used in BIMERR Urban Planning module

Name of the technology/library	Version	License
Apache Tomcat	9	Apache License 2.0 license
Angular	8	MIT License
Typescript	3.5.3	Apache License 2.0 license
NGRX Entity/Store	8.5.2	MIT-style License
LeafletJS	1.6.0	BSD-2-Clause
Java OpenJDK	11	GPLv2
Spring Boot	2.2.1	Apache License 2.0 license
Hibernate	5.4.8	LGPL 2.1
PostgreSQL	9.5	PostgreSQL License (similar to BSD/MIT)

2.4 ENERGY NETWORK VISUALIZATION

The power grid in the renovation project's vicinity is displayed on the map using sample data which was generated manually in QGIS[3] and imported as GeoJSON[4] into RenoDSS (see Figure 3). The user can turn the power-grid layer on and off using the switch button provided.

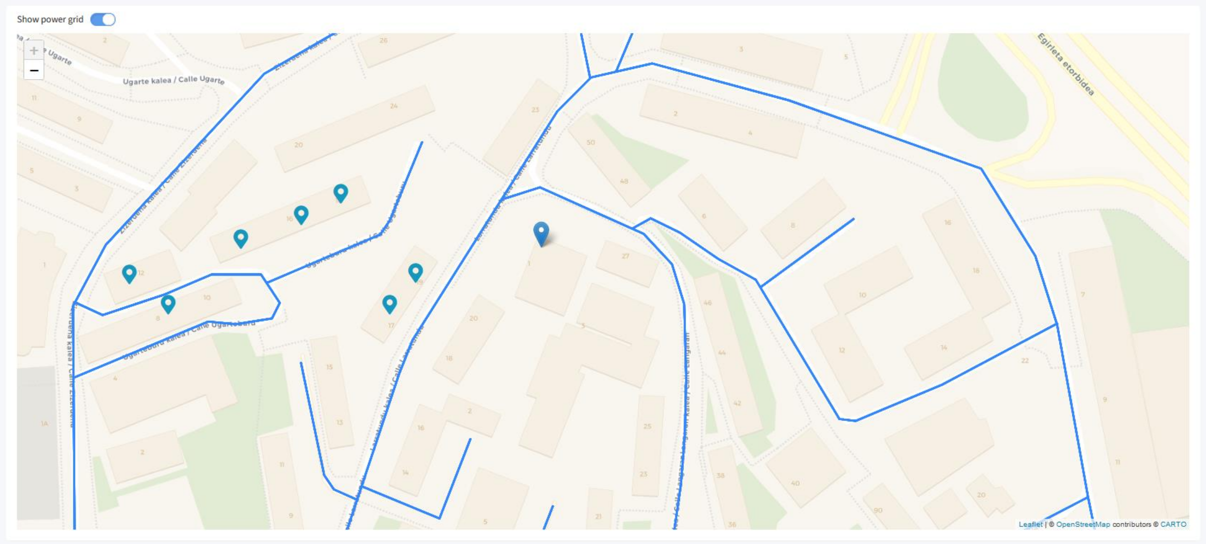

Figure 3: Show power grid

2.5 DISTRICT ENERGY EXCESS VISUALIZATION

The excess energy flow from/to the district (how much energy is consumed and fed back) is visualized on the map using the energy excess production profile calculation described in D7.7 – Section 2.8.

---

[3] https://www.qgis.org/en/site/, last access: 06.07.2021

[4] https://geojson.org/, last access: 06.07.2021

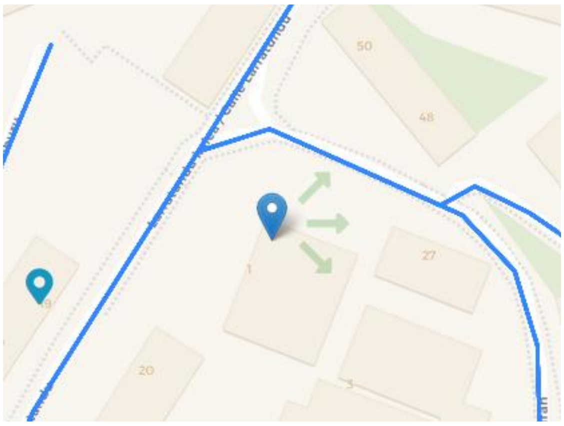

Figure 4: Excess production building

By clicking on the arrows from/to the district (see Figure 4) the calculated energy flow profile is shown as in Figure 5.

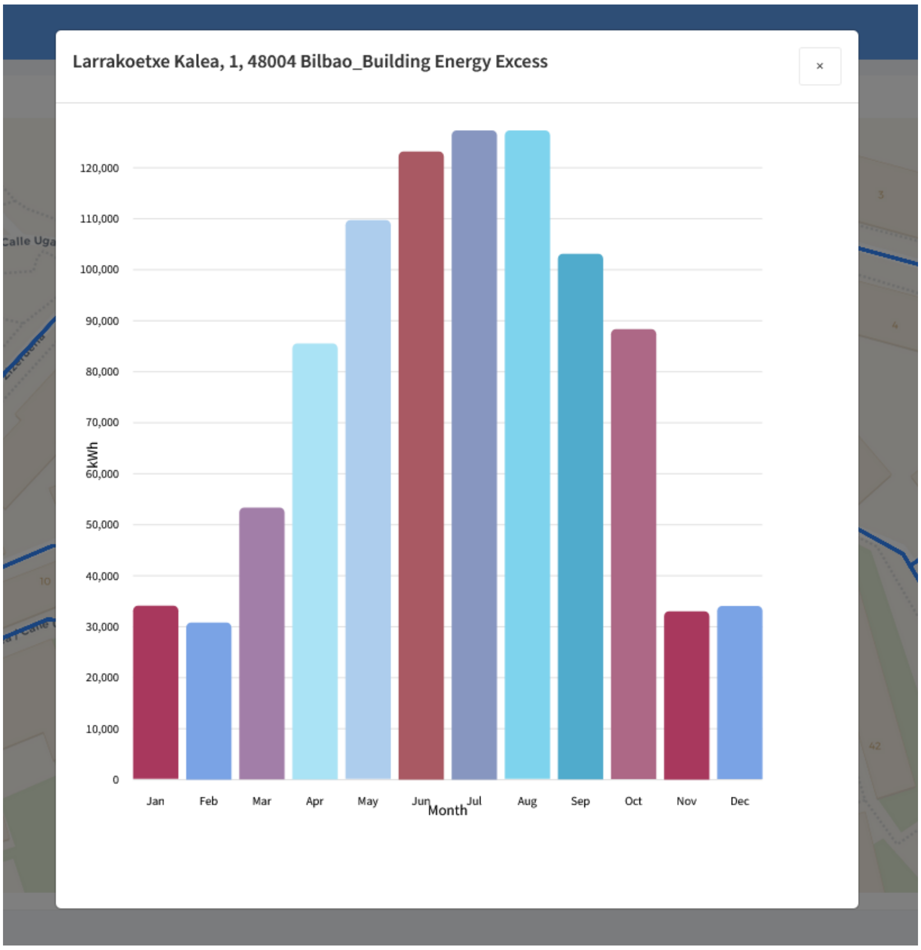

Figure 5: Excess production visualization

2.6 CityGML export

Functionality for converting building and network data into CityGML[5] data format is provided to the user (see Figure 6).

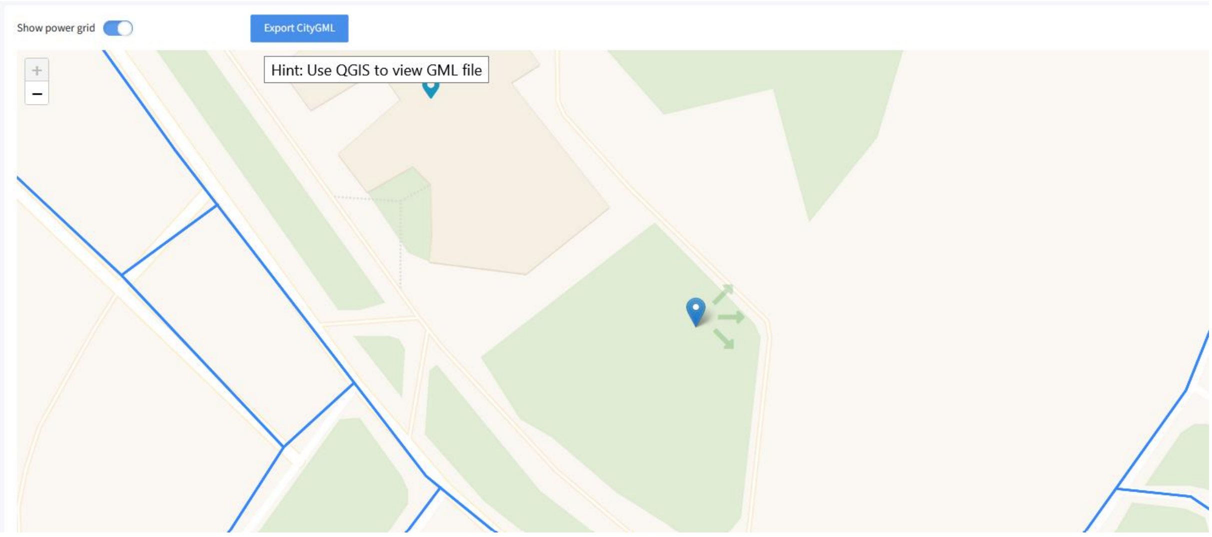

Figure 6: CityGML export functionality

When the user requests the CityGML export the corresponding IFC file is sent to the .NET open source xbim Toolkit[6] Service (running on RenoDSS server) where all relevant data for CityGML conversion is retrieved and sent back to RenoDSS. Data used for CityGML generation are location as well as the base geometry of the building along with direction vectors for coordinate conversion. IFC-file snippet displaying all relevant data:

```
#6=IFCCARTESIANPOINT((0.,0.,0.));

#110=IFCAXIS2PLACEMENT3D(#6,$,$);
#111=IFCDIRECTION((6.12303176911189E-17,1.));
#113=IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,1.E-05,#110,#111);
```

[5] https://www.lrg.tum.de/gis/projekte/citygml-30/, last access: 07.06.2021

[6] https://docs.xbim.net/, last access: 25.11.2020

```
#121=IFCGEOMETRICREPRESENTATIONSUBCONTEXT('FootPrint','Model',*,*,*,*,#113,$,.MODEL_VIEW.,$);

#206=IFCCARTESIANPOINT((37.16634,8.85925));
#208=IFCCARTESIANPOINT((57.65372,-16.21176));
#210=IFCCARTESIANPOINT((83.07197,1.08099));
#212=IFCCARTESIANPOINT((60.43167,22.67956));
#214=IFCCARTESIANPOINT((49.11152,16.01247));
#216=IFCCARTESIANPOINT((37.16634,8.85925));
#218=IFCPOLYLINE((#206,#208,#210,#212,#214,#216,#206));

#231=IFCSHAPEREPRESENTATION(#119,'Body','SurfaceModel',(#222));
#238=IFCSHAPEREPRESENTATION(#121,'FootPrint','Curve2D',(#218));
#241=IFCPRODUCTDEFINITIONSHAPE($,$,(#231,#238));
#247=IFCCARTESIANPOINT((-57.95617,0.,0.));
#249=IFCAXIS2PLACEMENT3D(#247,$,$);
#250=IFCLOCALPLACEMENT($,#249);
#251=IFCSITE('3_GOLIJ6n3NO89Dz8W9VFc',#42,'Surface:1040776',$,$,#250,#241,$,.ELEMENT.,(40,37,0,119
999),(22,56,59,999999),0.,$,$);
```

All geographical data is converted into Standard UTM coordinate system[7] and prepared so that the building can be properly displayed on a map. Each point defining a polygon is first converted into two 3D points using 0 and the buildings height extracted from the IFC-File as Z-values. The coordinates are converted from degrees-minutes-seconds to decimal degrees by the xBim-Service. This decimal degree representation has to be converted into Standard UTM. First the hemisphere and zone need to be calculated to be able to calculate easting and northing:

[7] https://en.wikipedia.org/wiki/Universal_Transverse_Mercator_coordinate_system, last access: 07.06.2021

```java
public static char getHemisphere(double lat) {
    if (lat <= 84 && lat >= 0)
        return 'N';
    return 'S';
}

public static int getZone(double lng) {
    return (int) Math.floor(lng / 6 + 31);
}
```

Figure 7: Calculating hemisphere and zone

The zone is then used to calculate easting:

```java
public static double getEasting(double lat, double lon, int zone) {
    double easting = 0.5
            * Math.log(
                    (1 + Math.cos(lat * Math.PI / 180)
                            * Math.sin(lon * Math.PI / 180 - (6 * zone - 183) * Math.PI / 180)) / (1
                                    - Math.cos(lat * Math.PI / 180) * Math.sin(lon * Math.PI / 180
                                            - (6 * zone - 183) * Math.PI / 180)))
            * 0.9996 * 6399593.62
            / Math.pow((1 + Math.pow(0.0820944379, 2) * Math.pow(Math.cos(lat * Math.PI / 180), 2)), 0.5)
            * (1 + Math.pow(0.0820944379, 2) / 2
                    * Math.pow((0.5 * Math.log((1 + Math.cos(lat * Math.PI / 180)
                            * Math.sin(lon * Math.PI / 180 - (6 * zone - 183) * Math.PI / 180))
                            / (1 - Math.cos(lat * Math.PI / 180)
                                    * Math.sin(lon * Math.PI / 180 - (6 * zone - 183) * Math.PI / 180)))),
                            2)
                    * Math.pow(Math.cos(lat * Math.PI / 180), 2) / 3)
            + 500000;
    easting = Math.round(easting * 100) * 0.01;
    return easting;
}
```

Figure 8: Coordinate conversion - easting

Zone and hemisphere are used to calculate northing:

```java
public static double getNorthing(double lat, double lon, int zone, char hemisphere) {
    double northing = (Math.atan(
            Math.tan(lat * Math.PI / 180) / Math.cos((lon * Math.PI / 180 - (6 * zone - 183) * Math.PI / 180)))
            - lat * Math.PI / 180) * 0.9996 * 6399593.625
            / Math.sqrt(1 + 0.006739496742 * Math.pow(Math.cos(lat * Math.PI / 180), 2)) * (1
                    + 0.006739496742 / 2
                            * Math.pow(0.5 * Math.log((1 + Math.cos(lat * Math.PI / 180)
                                    * Math.sin((lon * Math.PI / 180 - (6 * zone - 183) * Math.PI / 180)))
                                    / (1 - Math.cos(lat * Math.PI / 180)
                                            * Math.sin((lon * Math.PI / 180 - (6 * zone - 183) * Math.PI / 180)))),
                                    2)
                            * Math.pow(Math.cos(lat * Math.PI / 180), 2))
            + 0.9996 * 6399593.625 * (lat * Math.PI / 180
                    - 0.005054622556 * (lat * Math.PI / 180 + Math.sin(2 * lat * Math.PI / 180) / 2)
                    + 4.258201531e-05 * (3 * (lat * Math.PI / 180 + Math.sin(2 * lat * Math.PI / 180) / 2)
                            + Math.sin(2 * lat * Math.PI / 180) * Math.pow(Math.cos(lat * Math.PI / 180), 2)) / 4
                    - 1.674057895e-07 * (5 * (3 * (lat * Math.PI / 180 + Math.sin(2 * lat * Math.PI / 180) / 2)
                            + Math.sin(2 * lat * Math.PI / 180) * Math.pow(Math.cos(lat * Math.PI / 180), 2)) / 4
                            + Math.sin(2 * lat * Math.PI / 180) * Math.pow(Math.cos(lat * Math.PI / 180), 2)
                                    * Math.pow(Math.cos(lat * Math.PI / 180), 2))
                            / 3);
    if (hemisphere == 'S')
        northing = northing + 10000000;

    northing = Math.round(northing * 100) * 0.01;

    return northing;
}
```

Figure 9: Coordinate conversion – northing

Easting, Northing and the angle calculated using the direction vector provided in the IFC-file are then used to convert each point:

```java
double x = p.getX() * Math.cos(angle) - p.getY() * Math.sin(angle);
double y = p.getX() * Math.sin(angle) + p.getY() * Math.cos(angle);
p.setLocation(x + easting, y + northing);
```

Figure 10: Coordinate conversion - points

When all coordinates are transformed and converted each collection of coordinates is converted into objects for the GML representation:

```java
wallCoordinates.forEach(w -> {
    Polygon wall = geom.createPolygon(w, 3);
    wall.setId(UUID.randomUUID().toString());
    wall.setSrsName(srsName);
    surfaceMember.add(new SurfaceProperty(wall));
    surfaceMemberRef.add(new SurfaceProperty("#" + wall.getId()));
    AbstractSpaceBoundaryProperty createBoundarySurface = createBoundarySurface("WALL", wall, srsName);
    boundedBy.add(createBoundarySurface);
});
```

Figure 11: Example of CityGML Object generation

```xml
<boundary>
  <con:WallSurface>
    <gml:boundedBy>
      <gml:Envelope srsName="EPSG:25834">
        <gml:lowerCorner>669200.8881324463 4492558.354272831 0.0</gml:lowerCorner>
        <gml:upperCorner>669202.9489416634 4492561.113057955 6.65</gml:upperCorner>
      </gml:Envelope>
    </gml:boundedBy>
    <lod2MultiSurface>
      <gml:MultiSurface gml:id="d27d3dd6-c132-46e1-839b-3e18bedec2b4">
        <gml:surfaceMember>
          <gml:Polygon srsName="EPSG:25834" gml:id="433eb76c-8124-4153-87e4-f45da841e38a">
            <gml:exterior>
              <gml:LinearRing>
                <gml:posList srsDimension="3">669202.9489416634 4492561.113057955 0.0 669200.8881324463 4492558.3
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
      </gml:MultiSurface>
    </lod2MultiSurface>
  </con:WallSurface>
</boundary>
```

Figure 12: Wall representation in CityGML

The GeoJSON describing the buildings surrounding energy network (see Section 2.4) is parsed and translated to CityGML as well:

```java
FeatureCollection featureCollection = new ObjectMapper().readValue(new FileInputStream(network),
        FeatureCollection.class);

List<AbstractFeatureProperty> featureMembers = new ArrayList<>();

featureCollection.getFeatures().forEach(f -> {

    List<String> co = new ArrayList<>();
    MultiLineString multiLine = (MultiLineString) f.getGeometry();
    List<List<LngLatAlt>> coordinates = multiLine.getCoordinates();
    coordinates.forEach(l -> l.forEach(b -> {
        double x = CoordinateConverter.getEasting(b.getLatitude(), b.getLongitude(), zone);
        double y = CoordinateConverter.getNorthing(b.getLatitude(), b.getLongitude(), zone, hemisphere);
        co.add(x + "," + y);
    }));
});
```

Figure 13: Energy network conversion

The downloaded file can be opened using any available GML-Viewer, although QGIS is recommended to visualize all data on a map (see Figure 14).

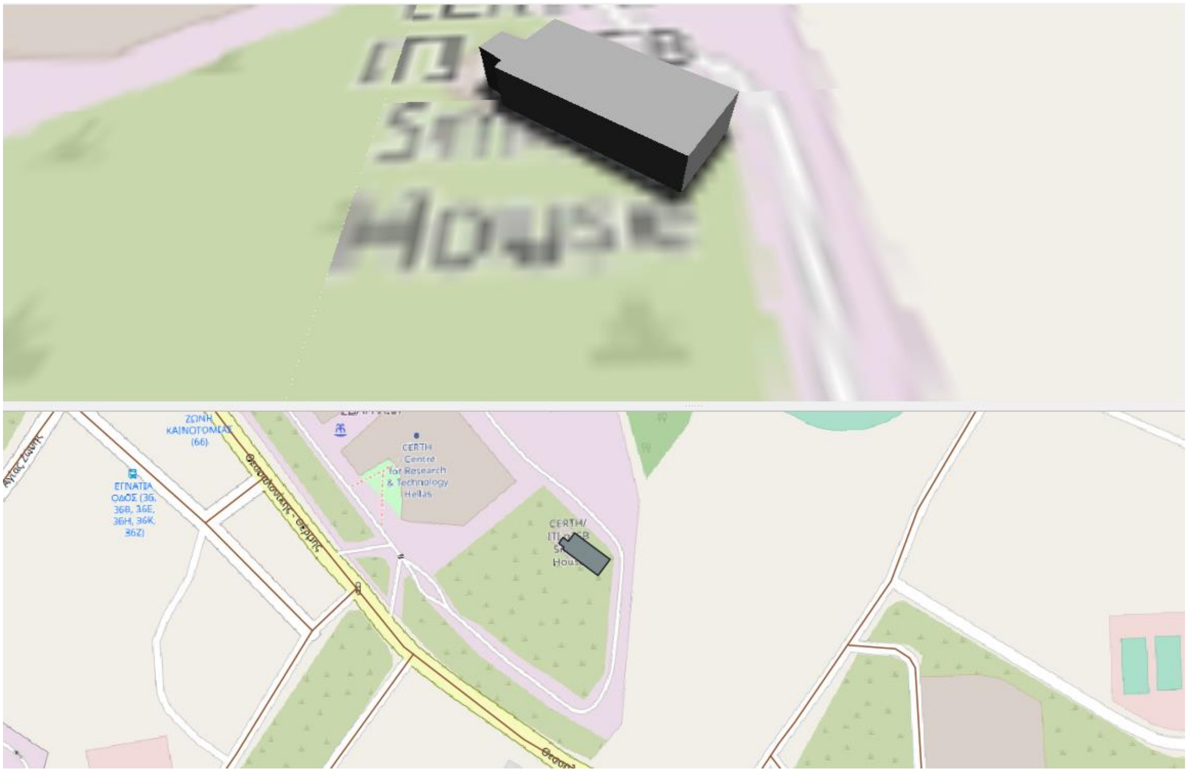

Figure 14: Exported CityGML file in QGIS showing building and energy networks in 3D and 2D perspective

2.7 WEATHER DATA RETRIEVAL

When the user opens a project in RenoDSS, the most recent weather data file is retrieved for calculating the energy KPIs. Weather data is automatically fetched via GET Request from EnergyPlus[8]. The returned JSON contains download links for weather data files for different locations in the world along with their coordinates. These coordinates are parsed to find the closest to the location of the renovation site. Then a second GET Request is sent using the download link provided to extract the EPW file which is then stored in RenoDSS along with the projects IFC and obXML files.

[8] https://raw.githubusercontent.com/NREL/EnergyPlus/develop/weather/master.geojson, last access: 10.05.2021

2.8 BUILDING VISUALIZATION

The current building and renovation scenarios are visualized based on the corresponding IFC files (based on third-party libraries).

All visualization components in RenoDSS are built on xbim WebUI and the generation of wexBIM files is necessary for displaying models in the frontend UI components. Whenever such a file is needed in the frontend, a request is sent to the backend. It is then checked if the requested file already exists. If it exists, then this file is returned to the frontend where it is displayed. If the file does not exist yet, the backend will send a request to the bimerr_xbim service that generates a valid wexBIM file from an IFC file. This file is returned to the frontend where the model is displayed.

2.8.1 Base Features

The 3D visualization supports basic functionalities such as the ability to rotate the building, by clicking and dragging, and a zoom functionality that can be used by scrolling the mouse wheel while hovering over the 3D view (see Figure 15).

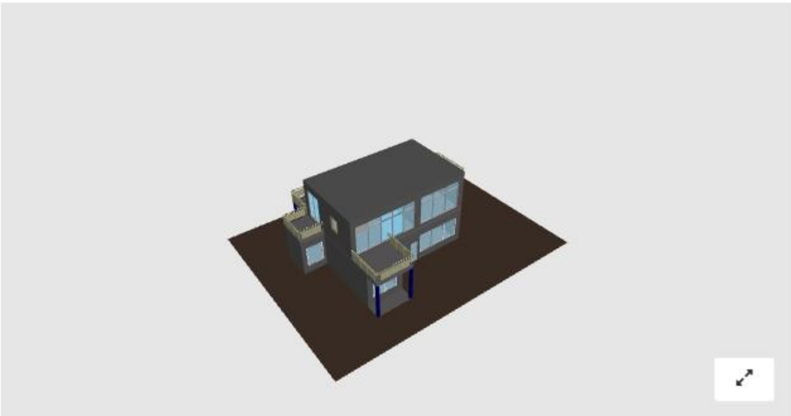

Figure 15: Visualization in base data view

2.8.2 Visualization modal

The visualization modal can be viewed by clicking the expand button in the standard visualization component. It serves the purpose of enlarging the 3D scene, and giving a better view on the model, depending on its context, it provides different functionalities (e.g., the visualization modal in the base-data screen will provide different functionality than the one in the measure screen).

2.8.3 Base-data visualization modal

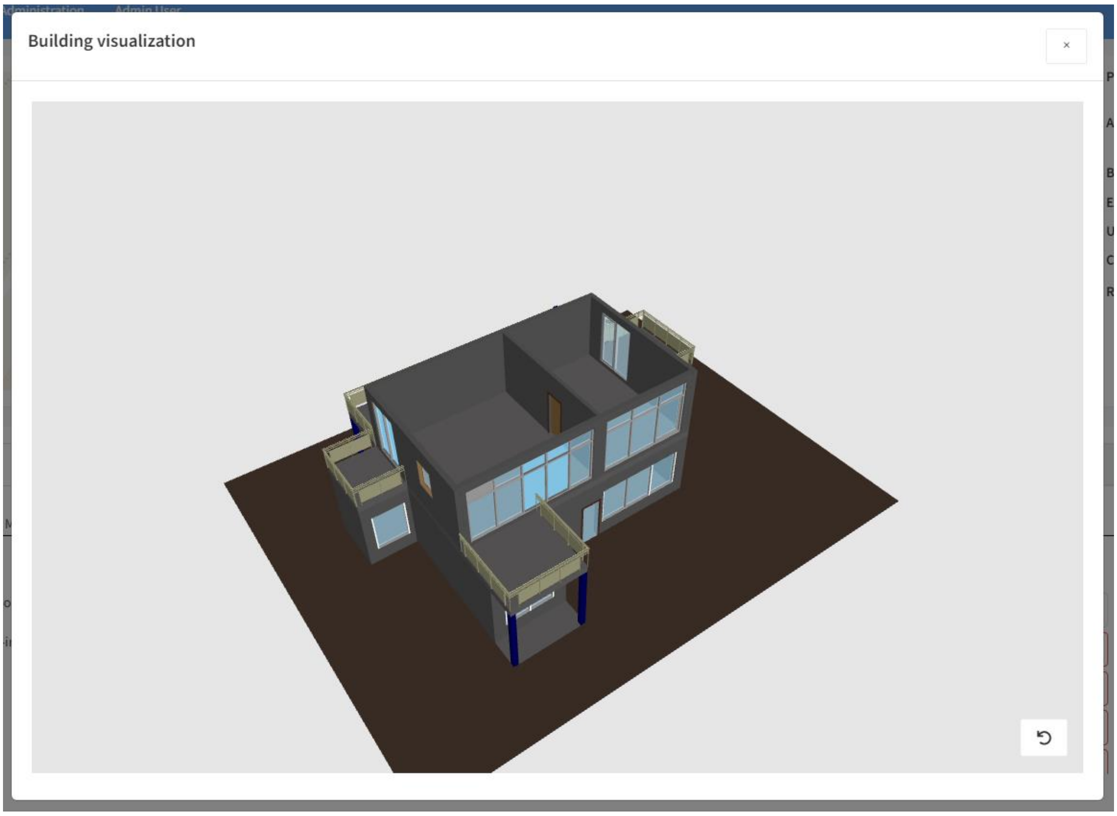

Figure 16: Model with hidden roof

To give better insights into the model this modal enables the user to click on building elements to hide them (see the example in Figure 16 with the hidden roof). This provides a view of elements that might otherwise be hidden deep inside the model. All elements can be unhidden by clicking the reset button in the lower right corner of the modal.

2.8.4 Measure visualization

The visualization in the measure screen is closely integrated with the list of IFC elements in the same view. Elements in the model displayed here are colored red if selected in the relevant list, light red if not selected, or gray if the element is not part of the list at all.

A selection in the list will trigger a color change of the element in the model, and clicking an element in the model, will select or deselect that element in the list, if it appears in it.

The modal implementation of this visualization is functionally the same, although larger for a better view of the model.

Figure 17: Model with selected façade measure

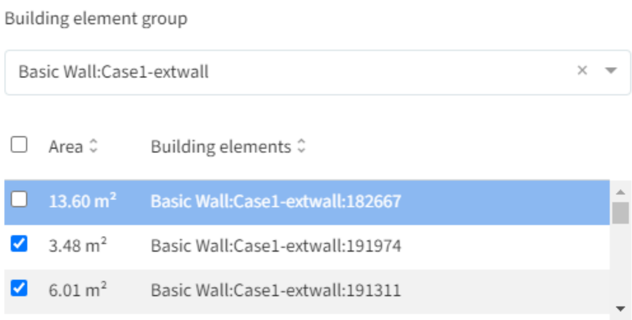

Figure 18: IFC Element list for the model

If the list of IFC elements changes, that is if another element group or building element group is selected, the model will update as well, setting element colors according to the new list.

2.8.5 Scenario visualization

The scenario visualization serves as a tool to visualize which elements are modified in a particular scenario. Here unaffected elements are colored gray, while modified ones are colored red.

The model changes when a scenario is selected and sets colors according to the selected scenario. The modal provides a larger view of the model and has the same hide/unhide functionality as the base-data visualization modal.

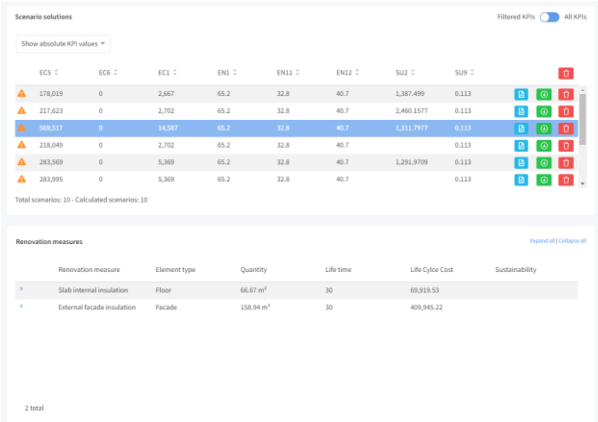

Figure 19: Scenario with roof and wall measures

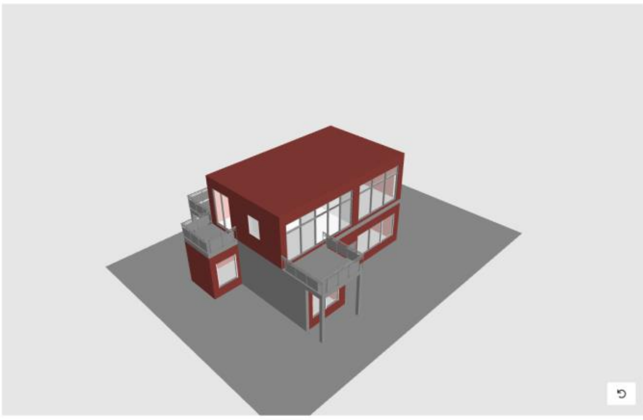

Figure 20: Model with roof and wall measures

2.9 MANAGEMENT OF IFC, OBXML, AND EPW FILES

When the user selects a project from the project list all corresponding files are loaded (see Figure 21).

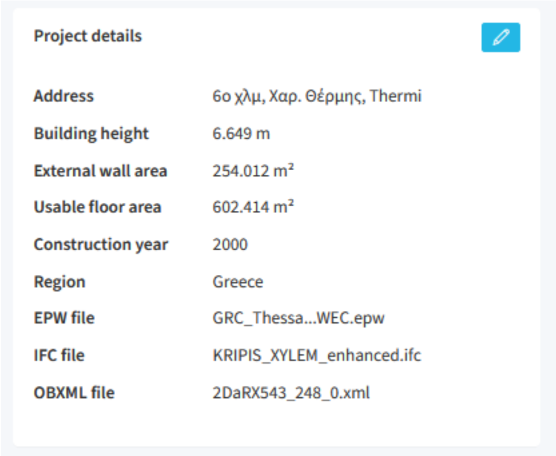

Figure 21: Project details in base data view

If one of the files (EPW, IFC, obXML) is missing, a dialog is shown where the user can upload the missing files. If no EPW file exists for the selected project, it is automatically downloaded from EnergyPlus (see Section 2.7).

The new modal dialog offers a manual refresh button to download the latest EPW file from EnergyPlus as well as functionality to download all files available for the selected project (see Figure 22).

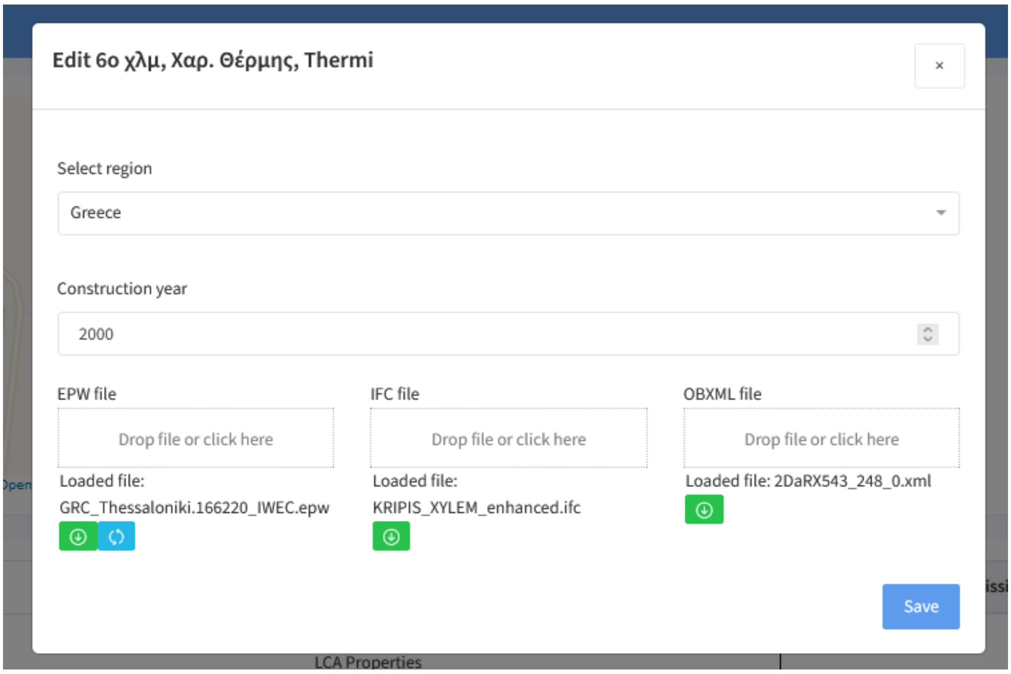

Figure 22: Modal for editing project information and uploading project files

2.10 API Documentation

The communication between the front end (RenoDSS UI) and the back end (Urban Planning module) is typically through internal APIs. As they serve inter-subcomponent integration purposes, they are not documented at this point in detail.

2.11 Assumptions and Restrictions

The final release of the BIMERR Urban Planning module is based on the following assumptions/restrictions:

- The current implementation assumes that calculation of excess energy production is restricted to electricity.
- The current implementation assumes that all default load profiles provided by the admin for a certain usage type and geographical location contain values in a 15-minute-resolution.
- The current implementation assumes that load profiles provided by the user for a certain building contain absolute values in kilowatt-hours in a 15-minute-resolution.

2.12 Installation Instructions

The BIMERR Urban Planning module is part of BIMERR RenoDSS, which is accessible via a web-based GUI, and therefore does not require installation or downloading of any component to use it.

2.13 Licensing

The KPI calculation schema is public. As parts of the BIMERR Urban Planning module are based on and integrated into the Xylem business intelligence platform it is a closed source component.

3. END-TO-END USAGE WALKTHROUGH TO THE BIMERR URBAN PLANNING MODULE

After selecting a renovation project in the project list on the dashboard, the user can navigate to the Urban Planning Module by selecting the button “Go to urban planning” on the bottom of the RenoDSS base data view (see Figure 23).

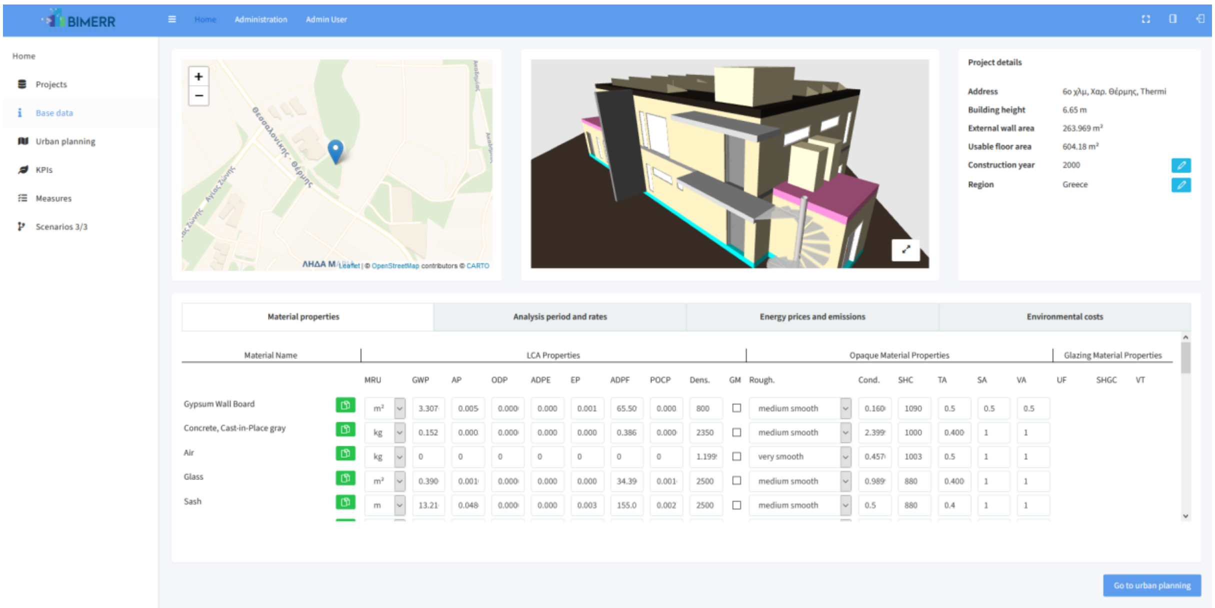

Figure 23: RenoDSS base data view

The Urban Planning view consists of a map showing the renovation project and surrounding buildings (see Figure 24).

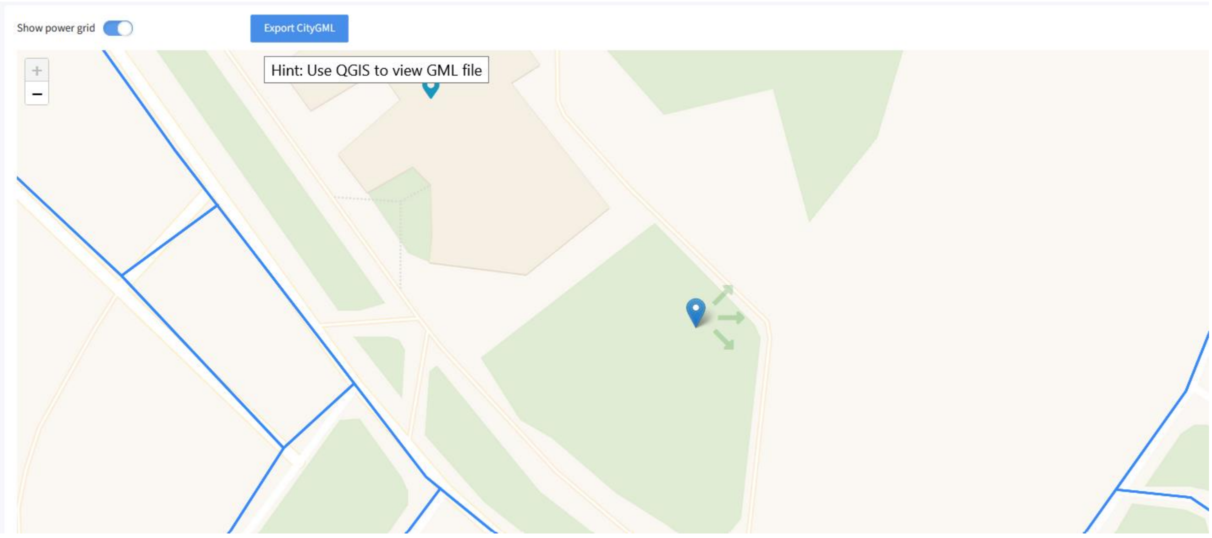

Figure 24: Urban Planning view

To create a new building the user can double-click the map on the location of the building to be created. Details on this process are described in D7.7. To show detailed information about a building or to edit the building's data the user can click any pin on the map to open a modal where all available data is shown and can be edited.

The user can turn the power-grid layer on and off using the switch button provided. The power grid in the renovation project's vicinity is displayed on the map using sample data which was generated manually in QGIS and imported as GeoJSON into RenoDSS.

The excess energy flow from/to the district (how much energy is consumed and fed back) is visualized on the map by green outgoing and red incoming arrows using the energy excess production profile calculation described in D7.7 – Section 2.8. By clicking on the arrows from/to the district the calculated energy flow profile is shown (see Section 2.5).

By clicking on the "Export CityGML" button the user can export a CityGML representation of the building and its surrounding energy networks. The CityGML file can be opened in a compatible viewer software (e.g., QGIS) as shown in Figure 25.

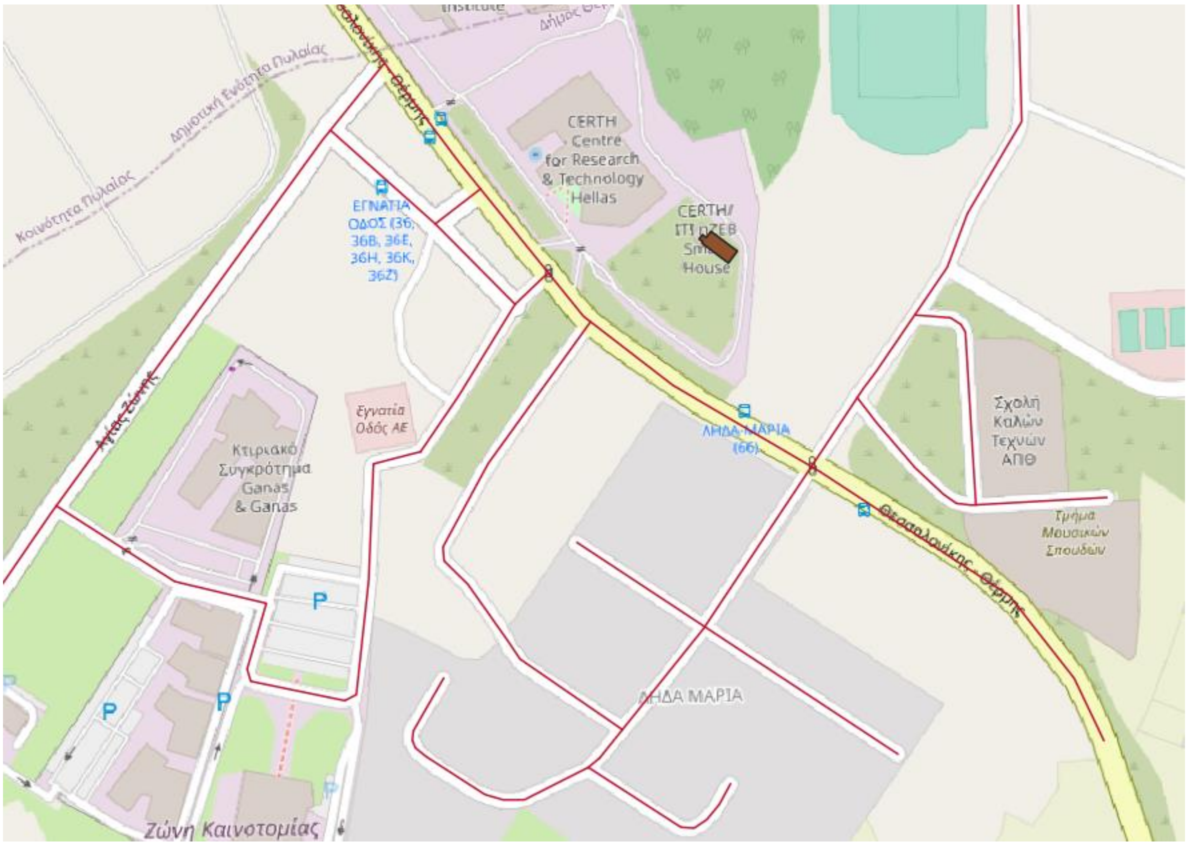

Figure 25: CityGML export

The KPIs calculated by the Urban Planning module are shown in the RenoDSS KPI view which can be opened by clicking on button “Go to KPIs”.

4. CONCLUSIONS

In its final version, the BIMERR Urban Planning module (i) provides the geographical perspective of the building under renovation and its surrounding buildings based on the geolocation extracted from the building's IFC file, (ii) allows to set energy production and consumption profiles of surrounding buildings via a map view, (iii) calculates the urban planning KPIs "EE13 – Electrical energy generated in the district and used onsite" and "EE14 - Energy generated on site and exported to the district", (iv) integrates energy network data and visualizes energy networks based on sample data, (v) provides CityGML export download functionality to ensure that available building and energy network data can be downloaded by the user in a standardized data format, (vi) visualizes excess energy flows on the district level, (vii) provides an automated weather data retrieval to support the building energy performance calculations with accurate weather data, and (viii) visualizes the current building and renovation scenarios based on the corresponding IFC files (based on third-party libraries).