VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
INSTITUTE OF COMPUTER SCIENCE
DEPARTMENT OF COMPUTATIONAL AND DATA MODELING

# Area's 5 Technical specification

Done by:

Viktar Breida, Petras Liubartas Rudys,
Jovilė Kiršytė, Shravani Deshpande,
Mindaugas Pumputis

Supervisors:
Virgilijus Krinickij,
Gediminas Rimša

Vilnius
2022

# Contents

# 1 Context diagram

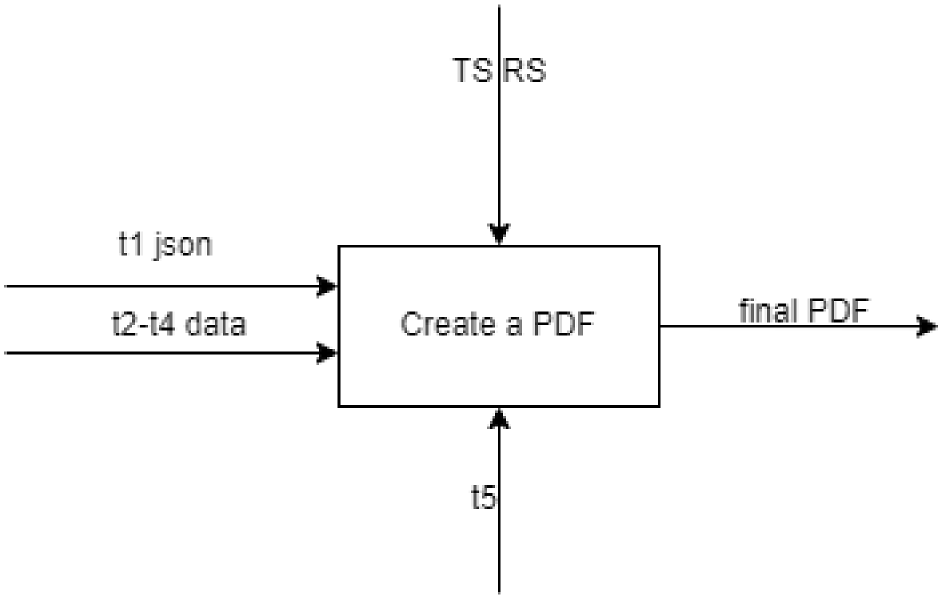

Figure 1. Context diagram.

We will use Technical specification and Requirement specification (TS and RS) to describe our project and use it for guidance We will get json from team 1 and data added to that json from teams 2 - 4 We will finalize the process by creating images and text and adding it to a pdf document.

## 2 UML deployment diagram

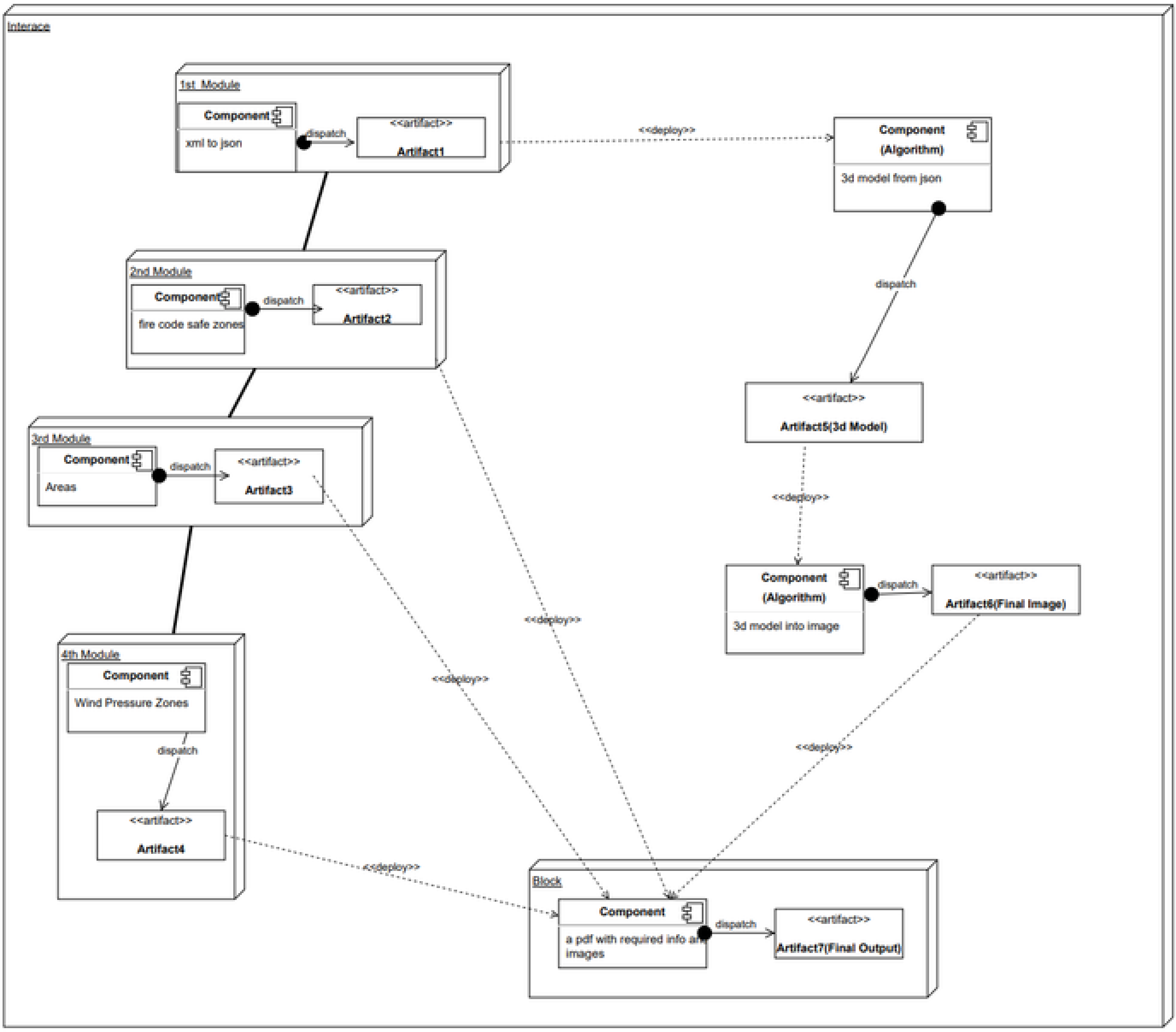

Figure 2. UML deployment diagram.

Interface includes conversion of xml to json which we got from team 1, then creating the artifact for fire code safe zones, areas of solar panel and wind pressure areas, then the algorithm for 3d models from json and adding it to pdf file. all of these will be deployed into our system. Finally the interface - to application.

# 3 Use case diagram

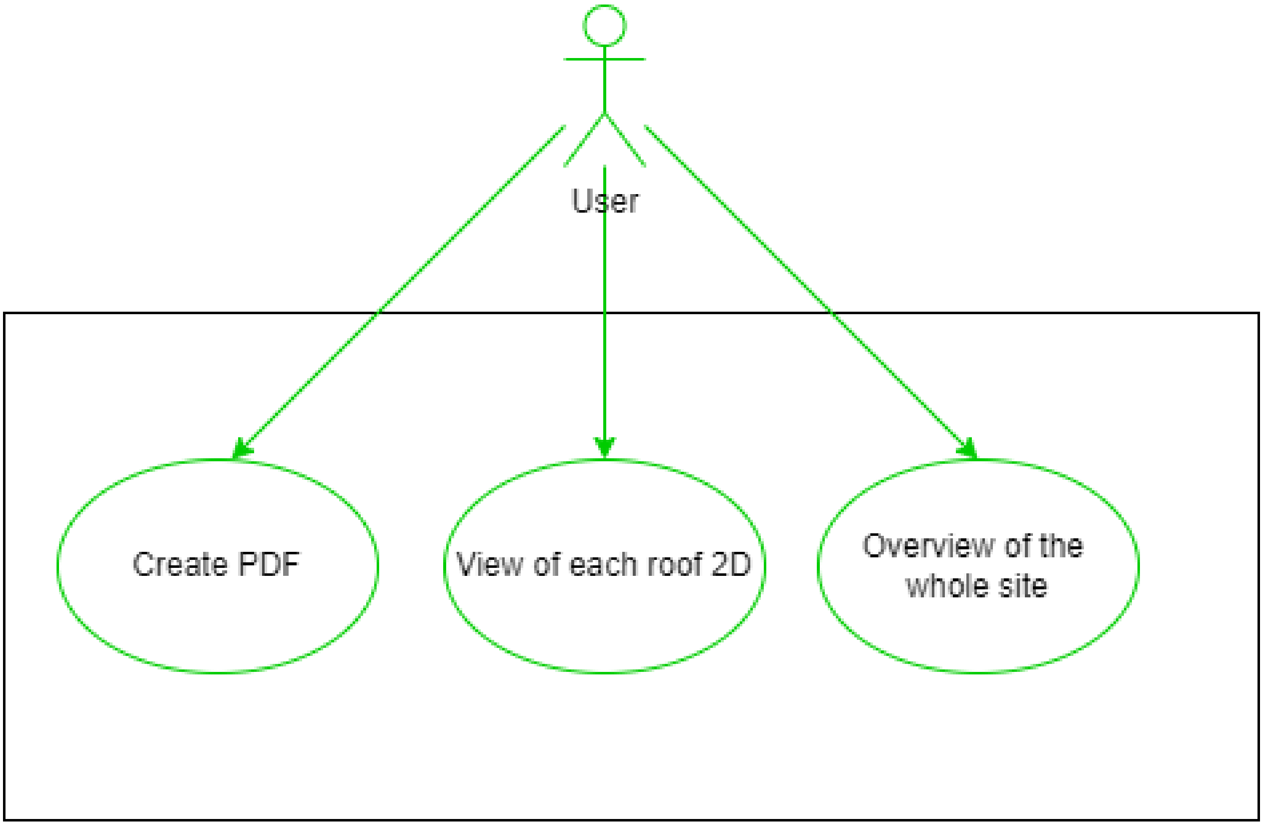

Figure 3. Use case diagram.

In our program user have several options:

1. Create PDF.
2. View of each roof 2D.
3. Overview of the whole site.

Application doesn’t need administrator and guest role.

# 4 Class diagram

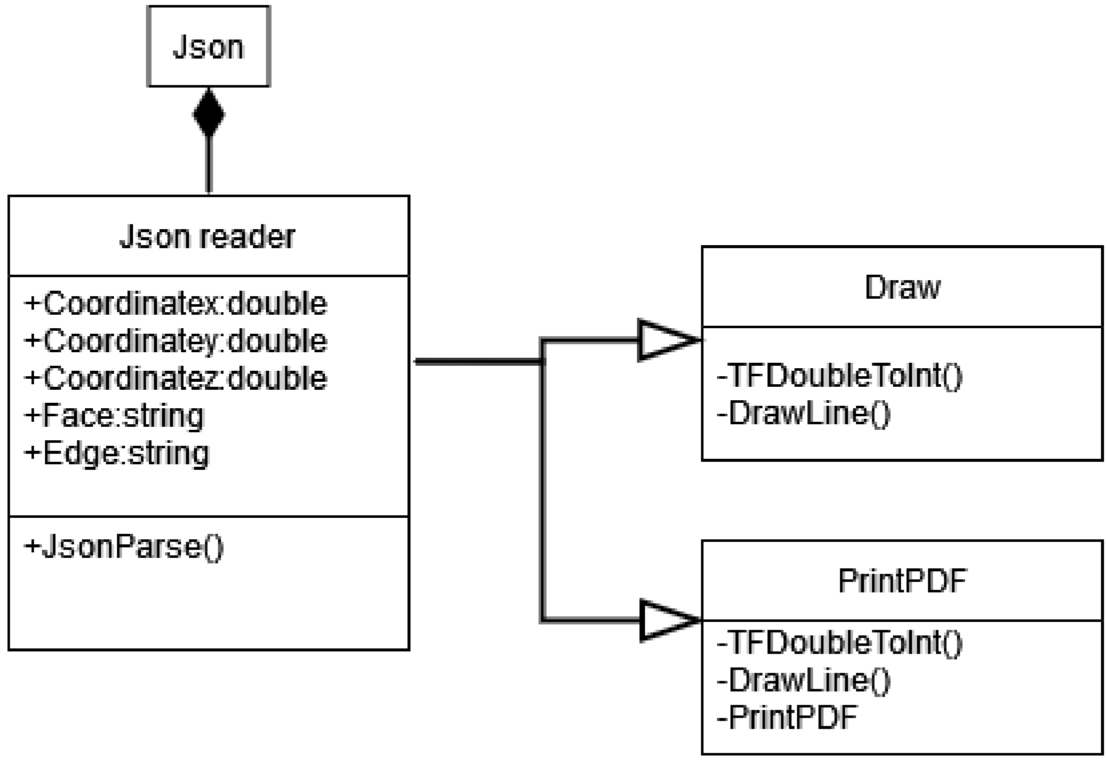

Figure 4. Class diagram.

In our diagram we have:
**JsonReader** - is a class that parses a json and gives data to ”Draw” and ”PrintPDF” classes.
**Draw** - is a class that basically draws the lines on a screen.
**PrintPDF** - is a class that gives out a finish PDF.

# 5 UI prototype

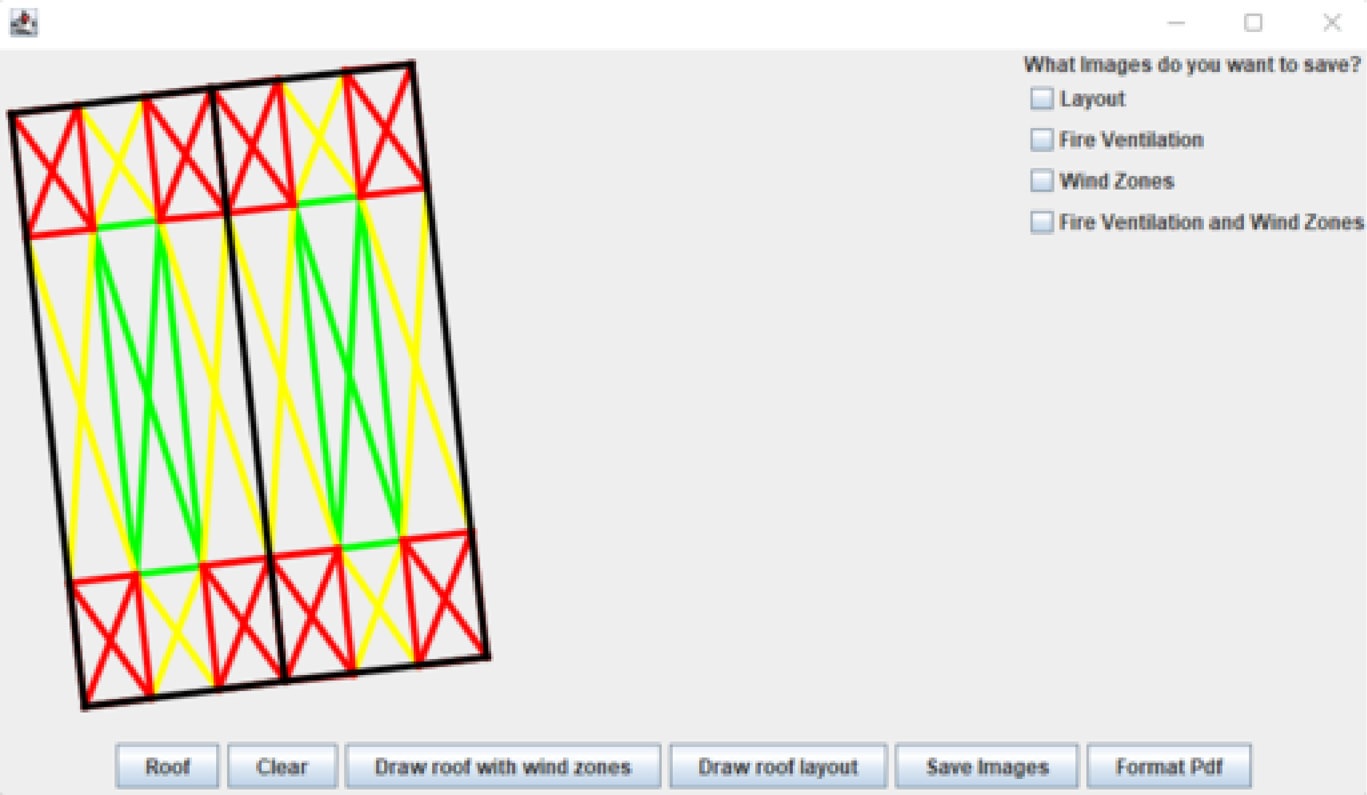
What Images do you want to save?
□ Layout
□ Fire Ventilation
□ Wind Zones
□ Fire Ventilation and Wind Zones
Roof
Clear
Draw roof with wind zones
Draw roof layout
Save Images
Format Pdf

Figure 5. UI prototype.

From Home page user can press 3 buttons:

1. **Roof** - it will draw overview of the whole site with fire ventilation setbacks.
2. **Clear** - this button will delete all drawings from window.
3. **Draw rood with wind zones** - this button draw overview of the whole site with wind pressure zones.
4. **Draw rood layout** - this button draw overview of the whole site with wind pressure zones.
5. **Draw rood with wind zones** - this button draw roof with wind pressure zones.
6. **Save Images** - this button will save selected images to project foulder.
3. **Format PDF** - this button creates PDF document with different views of the roofs.

The user can press ”Roof” button and ”Draw roof with wind zones” to get a combined image of the whole site overview with fire ventilation setbacks and wind pressure zones.

# 6 Structural aspects

1. Module that converts file from xml to json (provided by team 1).
2. Module that draws fire code safe zones (provided by team 2).
3. Module that draws the area where solar panels can be placed (provided by team 3).
4. Module that calculates wind pressure zones (provided by team 4).
5. Algorithm that draws 3d model from json.
    a. Json reader.
    b. Store json info in temporary file.
    c. Drawing algorithm.
6. Algorithm that converts 3d model into image.
    a. Draw image from predetermined points.
7. Algorithm that creates a pdf with required info and images using ”itextpdf” library.
    a. Create pdf.
    b. Add Text.
    c. Add Images.
8. User Interface – desktop application.

# 7 Dynamic aspects

After a user clicks a button in the desktop application:

1. The first module runs and converts file from xml to json
2. The second module runs draws fire code safe zones
3. The third module runs and draws the area where solar panels can be placed
4. The fourth module runs and calculates wind pressure zones
5. An algorithm runs that draws 3d model from json
6. Algorithm that converts 3d model into image
7. Algorithm that creates a pdf with required info and images.

# 8 Testing

**JUnit testing**
Java Unit Testing is basically testing the Java application at the time of production. It is a method of testing the source code for fitment of use in a production environment. Java Unit testing helps in testing individual components

# 9 Technologies and tools

- **Java programming language** – Every team agreed that we will use this programming language.
- **Github** – version control.
- **Eclipse IDE, IntelliJ IDEA IDE** – We will have to use both of these IDE's since some teams prefer one IDE over the other, but we have to work with everyone's parts of the project.
- **Junit** - for testing.
- **Discord, Teams** - communication.
- **TickTick** - work progress tracking.