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

Information Technology II year

Technical Specification

Software Engineering Project | Team 3

Done by:
Titas Majauskas
Dinas Majauskas
Jomantas Užusinas
Vilius Juknevičius
Sakalas Stasiulis

Supervisors:
Virgilijus Krinickij, Gediminas Rimša

Vilnius
2022

Contents

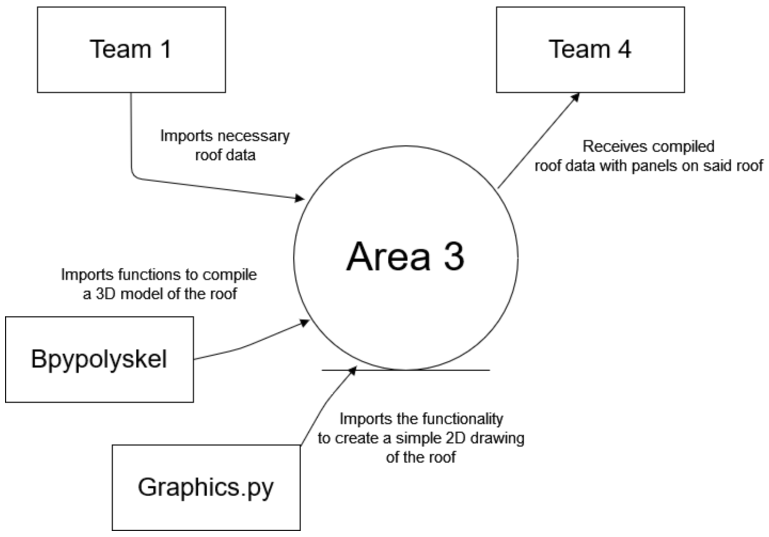
1 Purpose

1.1 Purpose of The Document

The purpose of this document is to provide an overview of the whole system, while visualizing and explaining the main parts of a system via diagrams, technological decisions and verbal explanations.

2 Diagrams

2.1 System-context diagram


Team 1
Team 4
Imports necessary roof data
Receives compiled roof data with panels on said roof
Area 3
Imports functions to compile a 3D model of the roof
Bpypolyskel
Imports the functionality to create a simple 2D drawing of the roof
Graphics.py

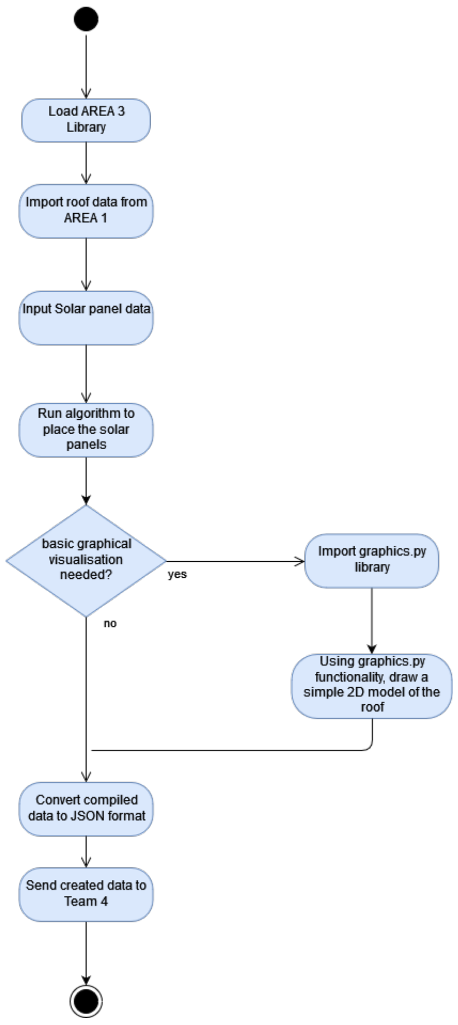
A **system context diagram**, shows **four** external objects that interact with the solution's library. These objects have either arrows pointing to them or from them.
Arrow meaning:

- An arrow line that points *from the system to an object* - shows what the **system provides** to that object.
- An arrow line that points *from an object to the system* - shows what that **object provides** to the system.

Objects meanings and purpose:

- **Team 1** - will only act as a provider to the system. It will supply Area 3 with the necessary data about the roof, most likely in JSON format as primary format style.
- **Bpypolyskel** - will only act as a provider to the system. It will supply the whole solution with functionalities needed to compile a 3D model of the given roof and provide a visualisation of that model.
- **Graphics.py** - will only act as a provider to the system. It will provide the system with functions to create a simple 2D model of said roof using basic data.
- **Team 4** - will act as a receiver of our library. Team 4 will receive the roof data that will be compiled through our algorithms. The format of the data sent to this team will be JSON as primary format.

2.2 Activity diagram


Load AREA 3 Library
Import roof data from AREA 1
Input Solar panel data
Run algorithm to place the solar panels
basic graphical visualisation needed?
yes
no
Import graphics.py library
Using graphics.py functionality, draw a simple 2D model of the roof
Convert compiled data to JSON format
Send created data to Team 4

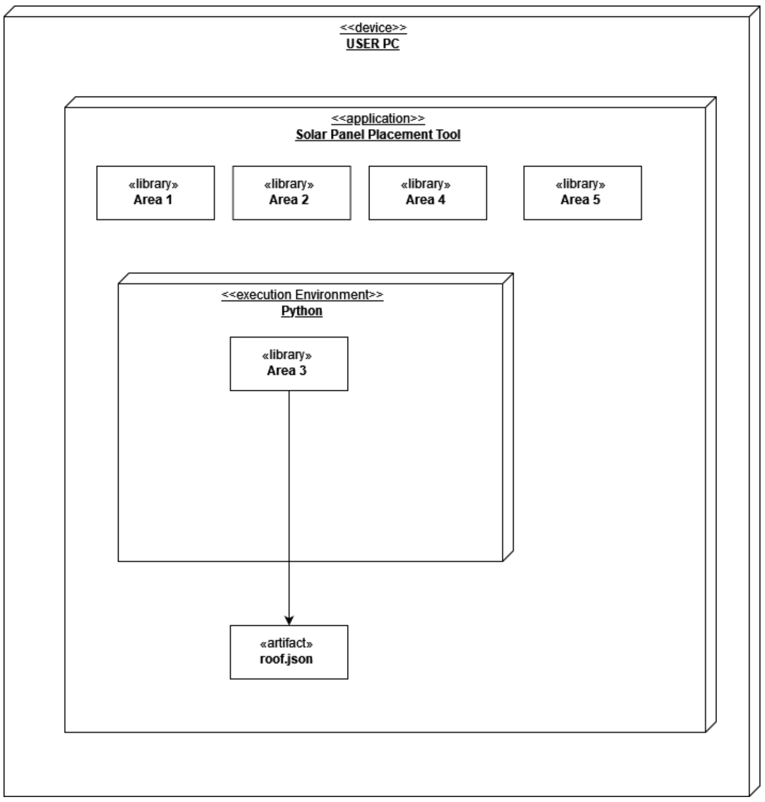
An **activity diagram** show the dynamic aspects of this solution. Our library provides **7** main activities, which work in coordination to provide our area's service. Furthermore, there will be **two** conditional activities, which will be executed if a condition is met. To receive the end result, we have to complete these steps:

- **Step 1** - having the library installed through *PIP*, we will import the library to our system using the default python import module. Example: *import solarPanelPlacer as spp*
- **Step 2** - after making our environment ready for data, we import the necessary roof data (roof polygons with their points), that will be extracted by Area 1. Primarily the data will be in JSON format and will be downloaded, imported into the working environment itself.
- **Step 3** - to finish up with data, last thing we have to import is the solar panel information, such as: gap between them, the measurements of the solar panels themselves.
- **Step 4** - we run the algorithms, that will be imported through our library, to make optimized calculations on solar panel placement and place the panels on the roof. (the panels will be represented as polygons on said roof)
- **Step 5** - check the necessity for basic drawing of the roof with panels.
- **Step 6** - if we need a visual representation, we continue as written below:

    - **Step 6.1** - import **graphics.py**, which will import the necessary functionality to draw a simple 2D roof.
    - **Step 6.2** - using the functionality of graphics.py, create a simple drawing of the roof with solar panels placed on it.

- **Step 7** - this step is executed after knowing, that no visual representation of the roof with solar panels is needed. The modifications done to the data of the roof should be converted back to JSON format and be compatible with other group members (Team 4 and Team 5). In addition the compiled data will be formed so, that later a 3D model of the roof could be created.
- **Step 8** - send the formatted data to Team 4. The data transaction will be simplified as - we will only add, modify the data of already existing JSON file, which will be accessible to the whole group from the launch of this system.

2.3 Deployment diagram



Deployment diagram represents how our solution will work internally in the system. The main device is the **user's personal computer**. It will run our main application called **Solar Panel Placement Tool** - this application will consist of **five** libraries and will use only one execution environment - Python. Having in mind, that we are talking about our area, the only executable library at that moment will be **Area's 3** library. The library will be responsible for receiving the roof's data, making optimized calculations on said data and placing the solar panels onto the roof where it is possible, meaning that the solar panels will not be set onto chimneys, skylights and similar objects that are meaningfully put on the roof. After needed calculations, the data will be formatted back to JSON format and will be sent to another team.

3 Technological Decisions

The library will be built using **Python** programming language in **Visual Studio Code IDE**. The library will be accessible to everyone through Python *preferred installer program* (PIP). Its code will use **Git** version control, and all commits will be archived in a designated repository.
For testing purposes our team will be using a Python testing framework called **Pytest**.

3.1 Visual Studio Code

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control.

3.2 Python

Python is a programming language used for web and software development, data analytics, machine learning, and even design. Python will be used to implement required solution. We will use Python 3.11.0 - the newest version available (2022-10-25).

3.3 Pytest/Pytest-Snapshot

- **Pytest** - a testing framework, which makes it easy to write small, readable tests, and can scale to support complex functional testing for applications and libraries.
- **Pytest-Snapshot** - a plugin for snaphshot testing with Pytest. Snapshot testing can be used to test that the value of an expression does not change unexpectedly.

3.4 Python preferred installer program

Python preferred installer program (PIP) is used for installing libraries for Python.

- **NumPy** - a Python library used for working with arrays. It also has functions for working in domain of linear algebra and matrices.
- **Matplotlib** - a comprehensive library for creating static, animated, and interactive visualizations in Python.
- **Mathutils** - a general math utilities library providing Matrix, Vector, Quaternion, Euler and Color classes.
- **Graphics.py** - a library used to draw simple mathematical figures, using two-dimensional vector points.
- **Bpypolyskel** - a library used to compile a 3D model of a roof.

4 Testing

Given the testing framework our team will use, testing will be done on these said sections of our solution:

- **Position:**
  - Do testing to check if the position of the solar panel will not be crossed with position of chimneys, skylights, etc.
  - Do testing to check if solar panels are not put over the roof's borders.
- **Coverage**
  - Do testing to check if solar panel covers the maximum possible area of the roof having in mind many limitations.

For later, more complex testing **Pytest** plugin **Pytest-Snapshot** will be used to check if some changes happened while compiling the needed data.

5 Live version

Overleaf read-only version - *click here*
Github repository for this documentation - click here