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

Software engineering | Information Technology 2nd year, 3rd group | Area 4

Technical Specification

Done by:
Arnas Černiauskas
Dovydas Martynėnas
Edvinas Raveikis
Oskaras Žukauskas

Supervisors:
Lekt. Virgilijus Krinickij
Lekt. Gediminas Rimša

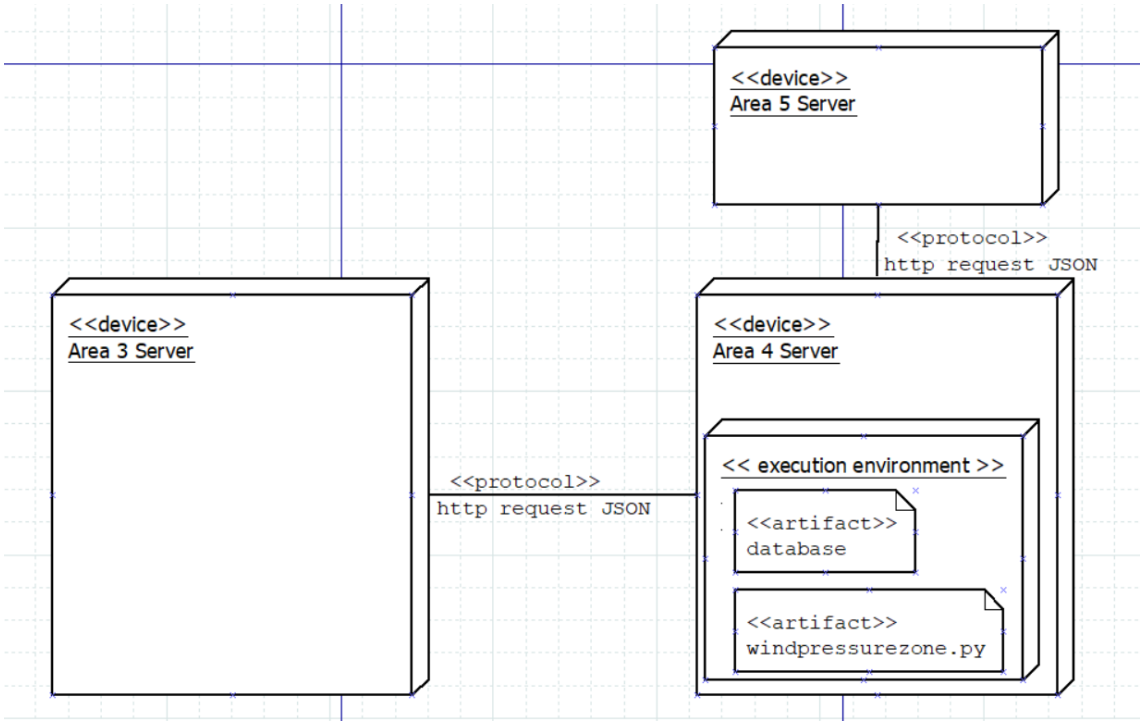
Vilnius
2022

Contents

1. Overview

There is no current system that would help a person to know and decide where to put solar panels on the roof of his house and what benefits or drawbacks would be and whether the solar panels would be blown off the roof by wind or not. The solution would be to make a program that would be able to calculate the wind pressure zones on the roof and help the customer decide whether the location he chose is all right or not and what wind pressure they can expect.

2. Deployment diagram



3. Goals

1. A roof type selection system
2. Solar panel placer
3. Wind pressure zone calculator

4. Product requirements

- The software product should be able to allow the user to select between multiple roof types and show where solar panels could potentially be placed based on the shape of the roof.
- The solar system engineers installing the solar panels need to know which mounting products to choose (i.e., attach solar panels using stronger parts where wind forces are higher). This decision is informed by dividing the roof face area into regions representing different relative wind strength. These regions are called “wind pressure zones.”
- Create an algorithm which calculates wind pressure zones in Python preferably using “bpy-polyskel” library using formula: a = 0.4 * mean roof height, but not less than 3 feet

5. Out of scope

- This program will not be able to generate new roof types, nor will it allow to use a custom roof that is not already in the database of existing roof types.
- This program will only make the required calculations. The algorithm will not compile when information is provided in letters, nor will it work if you provide wrong data that calculates into a non-logical number.
- Software will not be able to predict wind direction or how the weather will be in the future.
- This software is not responsible for weather disasters that can/will happen in the future and what implications they will have on solar panels.

6. Approach

Firstly, we will work with Team 3 which will obtain mandatory data for our algorithm that will calculate wind pressure zones. Then we will work on our algorithm to make sure it compiles correctly without any bugs or errors. Finally, we will need to work with Team 5 to help them develop the front-end of this final software so that it looks and works as intended.

7. Measuring impact

We will compare randomly generated data to the actual data to make sure the algorithm is consistent and can calculate given data. Additionally, we want to conduct an open questionary for Team 4,5 to evaluate their satisfaction with the product and identify problems with our algorithm.

8. Timeline

- Version 1 - The program acquire the JSON file for the input and parses that file
- Version 2 - The program calculates data using the mathematical formula: “a = 0.4 * mean roof height, but not less than 3 feet” by using the data provided by the user.
- Version 3 - The program calculates data using the JSON file
- Version 4 - The software algorithm is divided into input, function call and output to ensure quality syntax-wise.
- Version 5 - Bug fixes
- Version 6 - The output is added to JSON file for further use
- Version 7 - The program presents the final data to Area 5

9. Open Questions

- How will we split the work load during the software programming part?
- Will the user be able to understand how to use the program?
- How similar will Team 3s data be to our own mock-up test data?