



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

Solar system area 4 2022 TS

Technical specification

Done by:

Augustinas Jažauskas, Arnas Čekanauskas,
Gytis Sabaliauskas, Mažvydas Poviliūnas

Supervisors:

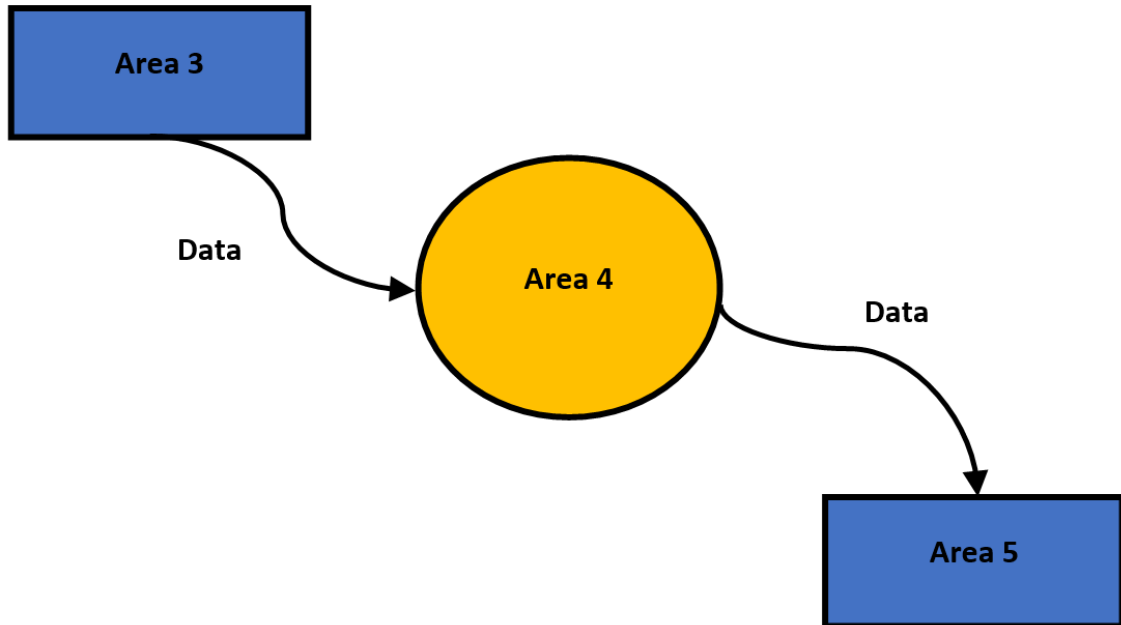
Virgilijus Krinickij,
Gediminas Rimša

Vilnius
2022

Contents

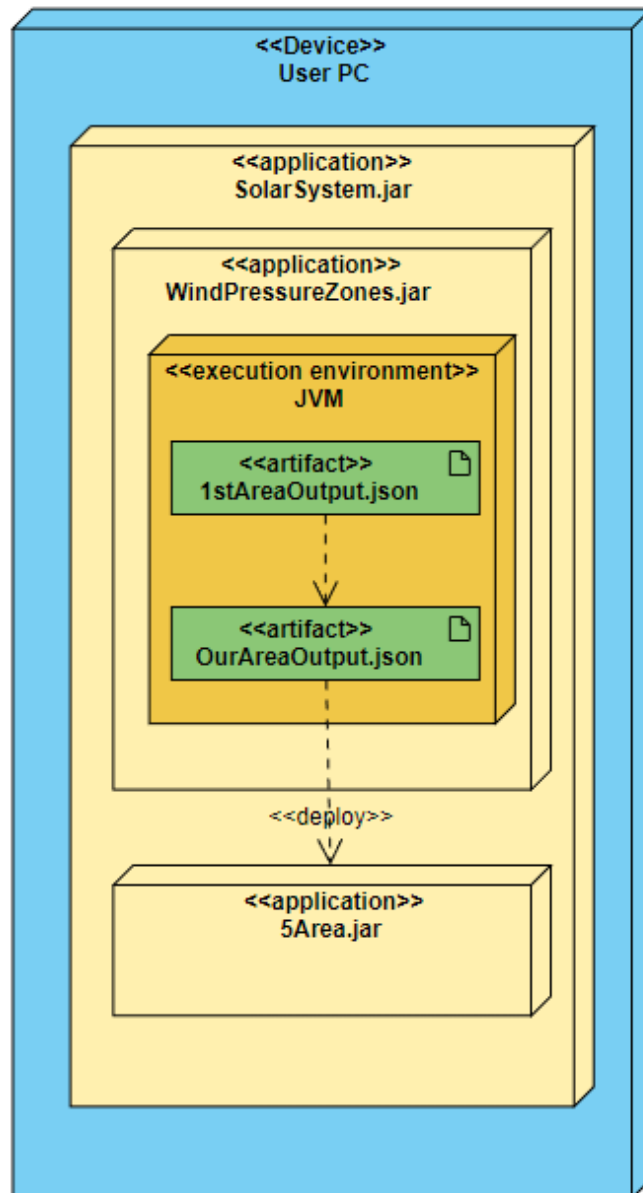
1	Context diagram	3
2	UML deployment diagram	4
3	UML class diagram	5
4	UML activity diagram	6
5	Structural aspects	7
6	Dynamic aspects	7
7	An overview of testing	8
8	Technologies and tools	9

1 Context diagram

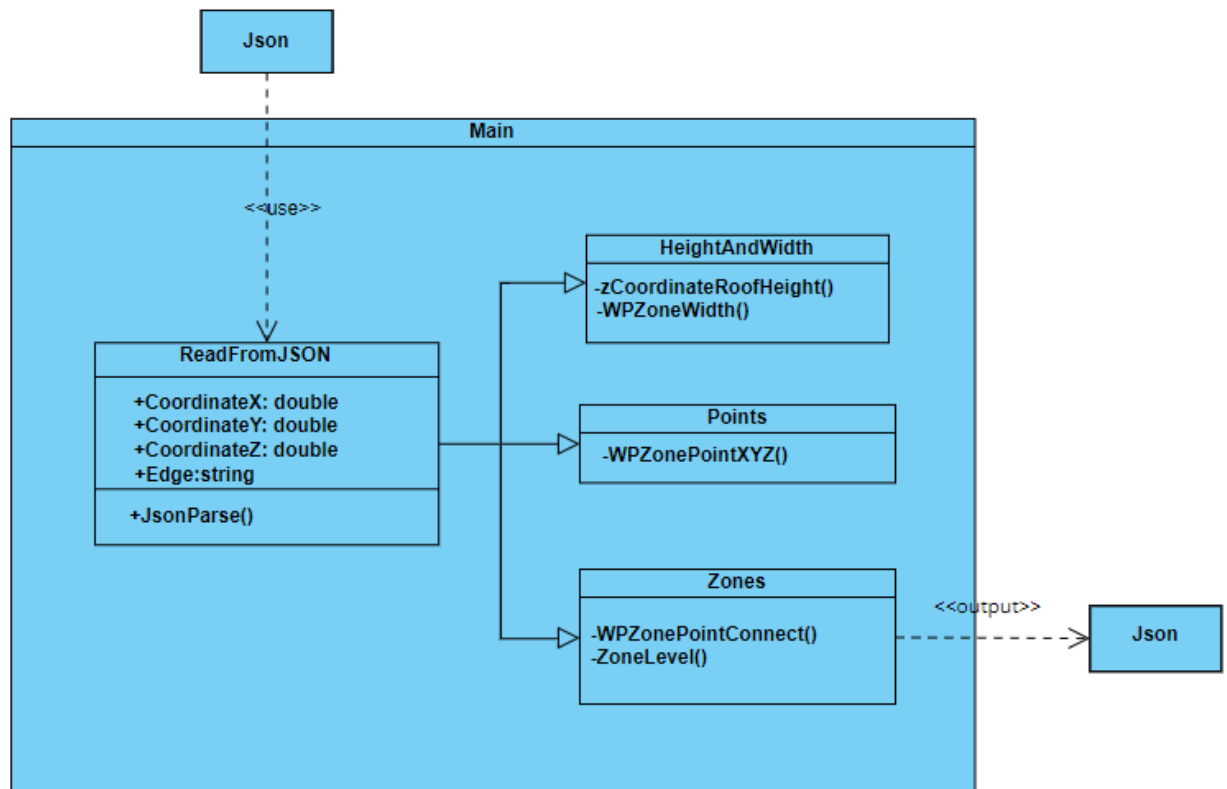


- Area 4 will receive a JSON file with the required data from area 3, after which area 4 will perform their task and send the output to area 5.

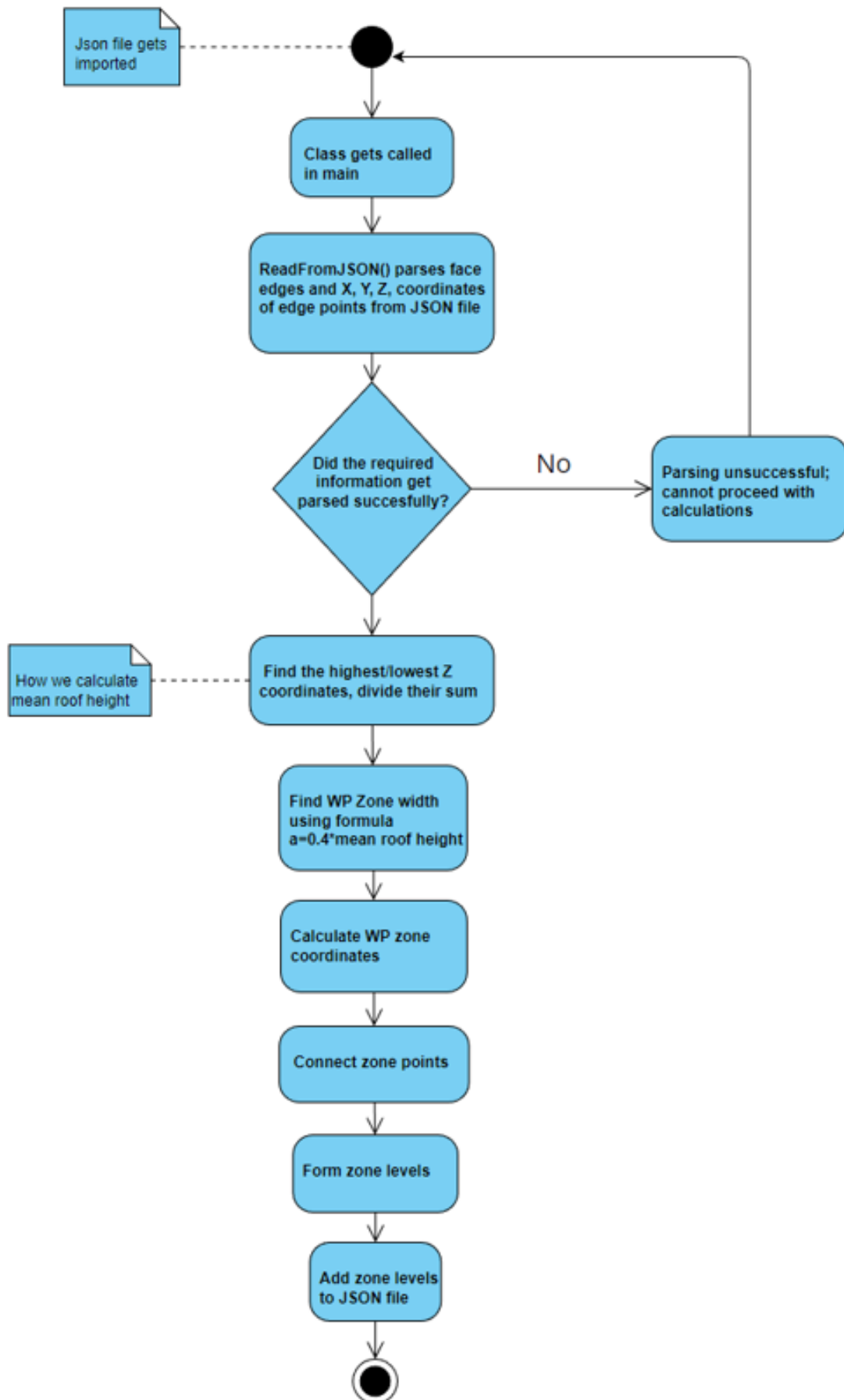
2 UML deployment diagram



3 UML class diagram



4 UML activity diagram



5 Structural aspects

The essential components will be:

- Main – connects and calls every class.
- ReadFromJSON – a class that reads the house roof edges and their point coordinates from the JSON file.
- HeightAndWidth – a calculation class that calculates the mean roof height by finding the highest and the lowest Z coordinate of the roof and dividing their sum, also finds the WP (Wind Pressure) zone width using the formula:
$$a = 0.4 * \text{mean roof height (if the result is less than 3, } a = 3).$$
- Points - the most crucial calculation class, which calculates the coordinates of every WP zone point.
- Zones - a class that connects the WP zone points, forms zone levels and adds them to the JSON file.

6 Dynamic aspects

- The JSON file - this component will depend and vary based on the client's use.
- The data from the JSON will be read and used in the software's classes to calculate the WP zone coordinates.
- The modified JSON with the calculated WP zones will be sent as output to area 5.

7 An overview of testing

All testing of our software will be automated, since our area's job is only to read the provided JSON, calculate WP zones and include them into the JSON file.

Which parts will be tested:

- Data reading from the JSON functionality.
- Height and width calculation functionality.
- Point coordinate calculation functionality.
- Zone level forming functionality.

Our approach of testing will be - code first, test after, since that it is the simpler method for our team.

8 Technologies and tools

1. The programming language will be Java, since the majority of the project members have the most experience working with it.
2. The coding will be done using IntelliJ, as it is one of the most useful and well known IDEs for JAVA.
3. Testing will be implemented using JUnit, because it is highly efficient and is supported by IntelliJ.
4. The code and documentation of the software will be repositied in a shared GitHub project repository, so that it would be more convenient to store data and collaborate with other teams, working on this project.
5. To improve planning, distribute tasks and track the work progress, the project will be managed in JIRA.
6. The documentation will be prepared using LaTeX, since that will allow us to make use of the given template and prepare a more proper document.
7. The communication tool used for the project will be Discord, because it is extremely simple to use and has great communication management features.
8. Since the overall project will be coded using JAVA, our software will work as a part of a Java application.
9. The software will be supported on Windows/Linux, because the majority of computer users use these computer operating systems.
10. As per standard practice, the software will run on a computer, since the client did not specify on what device it should work.