VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
INSTITUTE OF COMPUTER SCIENCE
«DEPARTMENT OF COMPUTATIONAL AND DATA MODELING» OR
«CYBERSECURITY LABORATORY»

Process of purchasing a solar system (at a high level) - Area 2

Technical Specification

Especificación Técnica

Done by:

Nicolás Simarro

Yassin Aomar

Martín Martínez

Miguel Rodríguez

Vilnius
2022

Contents

Introduction

The purpose of this document is to show the technical specification of our project including diagrams, a high-level overview of our team’s deliverable’s internals and an explanation of what technologies and tools that will be used to implement the team’s deliverable

1 High-level overview of our deliverable's internals

1.1 Structural:

To get all the useful Information and convert it to a json extension where our Project mates can work with, we have the following modules/parts:

- We have 3 Converters that are used for the following purposes:
  - A converter that gets all the Information from the XML and converts it into a 3dmodel to ¨build¨ the exact aspects of the house
  - A converter that extracts all the Information from the XML to a DTO model
  - A converter that converts the DTO Information to a JSON that our mates can work with
- We have a class that creates a database with all the Information that Will be used to create the different objects used to create the house with its roof.
- We have some different classes that are used to separate all the Information creating all the objects with the correct Information. One example of this classes is 'Location.java' where we can find the coordinates, address, orientation, etc.
- We have an error control that launches an exception if there is any failure with the Information that is being created. This module is very useful not to overload a server with failed calls to methods.
- We have a main part where all the previous modules work together, in this module we use all the converters to get the 3D model and the JSON from a XML.

1.2 Dynamic:

In our case when all the components work together we can find a program with an error handler, that extracts all the useful Information, creating a database, and converts all this information into a better format (json, 3dmodel) that will be sent to the next project group, with this format, they can analyce all the Information to get the optimiced way of putting solar panels in every different type of roof.

2 Technologies and tools that will be used to implement the team's deliverable

We have used some diferent tools and programs to get all our work done, we can separate it in diferent categories:

- To make all the diferent diagrams as the context or the UML deployment diagram we have used 2 diferent programs:
  - Visual paradigm
  - StarUML

  They have the same purpose but with some different nuances, therefore, we had diferent options to make it.

- To capture our ideas and make the code we used 2 diferent platforms as Eclipse or Visual Studio Code and the languaje we code with is Java because we all agreed was the one that could be easier for us to get our work done.

- In the case of the data, we work with different types like XML as the starting Data format that Will be converted to DTO format, where we can work on it to transform it to a json format, which is our goal.

- To make our document we used Word as starting platform where all the ideas and Information that we got from a brainstorm, that lately we would select or delete, use and transform it to the final format using LaTeX, as the best type of deliver documents.

3 Testing

3.1 Unit Testing

To test our product, we will set up a unit test framework where we will perform tests to check the correct functioning of the parts of our product and eliminate possible errors and side effects. To do so we are going to add checkpoints that allow us to manipulate the state and add observation points that allow us to see the inner workings of the service. Possible test that we are going to do is check if there is any failure with the information that is being created by the functions of the converters (from XML to DTO and from DTO to JSON). And once we are sure that we have obtained the information correctly, we will take advantage of unit tests to explore expected and unexpected scenarios to ensure the quality and stability of the code base as they are more efficient and effective than manual validations. We are going to add unit tests on each function to individually check the functions and methods created. We will probably use JUnit or Maven for the unit tests.

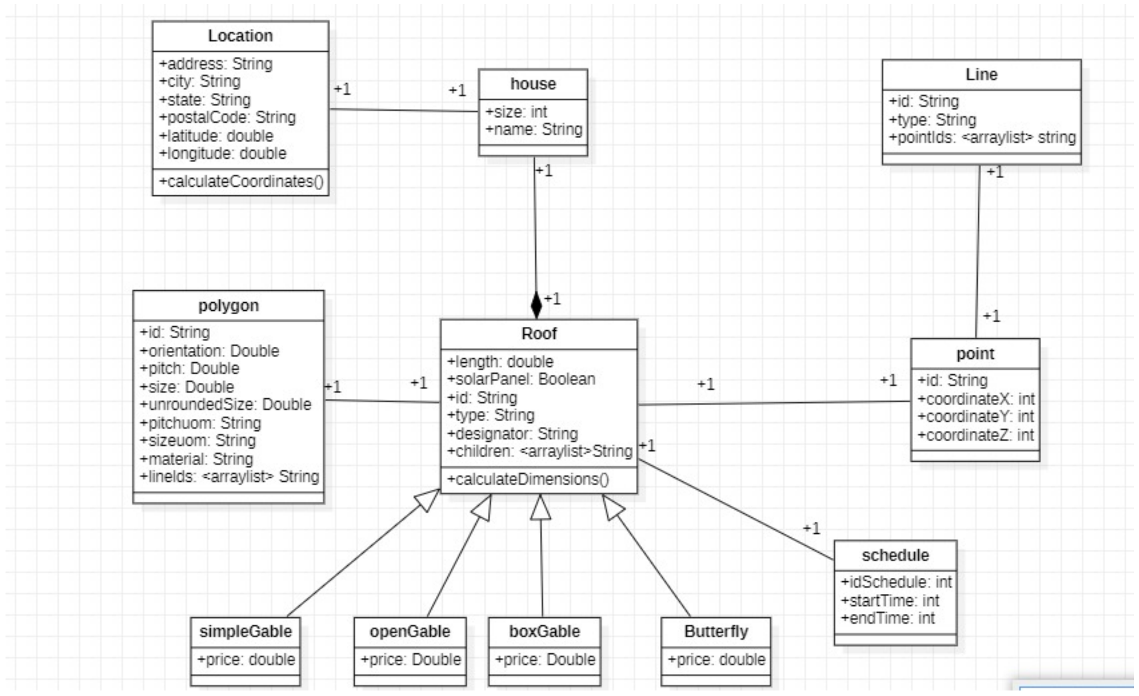
3.2 Integration tests

After the unit tests we will perform integration tests to verify that the different modules and services used by our application work in harmony when they work together (e.g, if the converters communicate properly). For that we are thinking about using Mocha or some similar test framework.

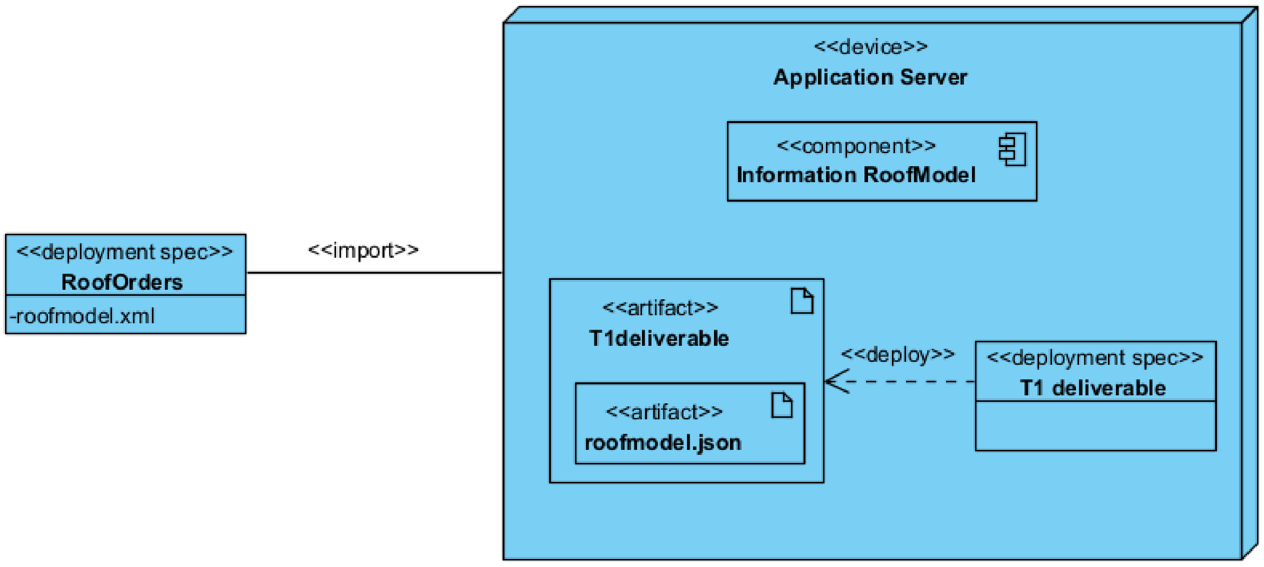
3.3 Performance tests

And to check the performance of our product we will perform smoke tests to verify the basic functionality of an application. These will be quick tests to run to make sure that the most important features of the system work as expected at the end that all the lines, dots and faces of the roof are joined together and to confirm that we have extracted the information correctly. So, if a smoke test fails, it means that there is a serious problem with the functionality of our software. Then, we should not deploy new changes until the failures are addressed. And if they fail in production, fixing them will have the highest priority.

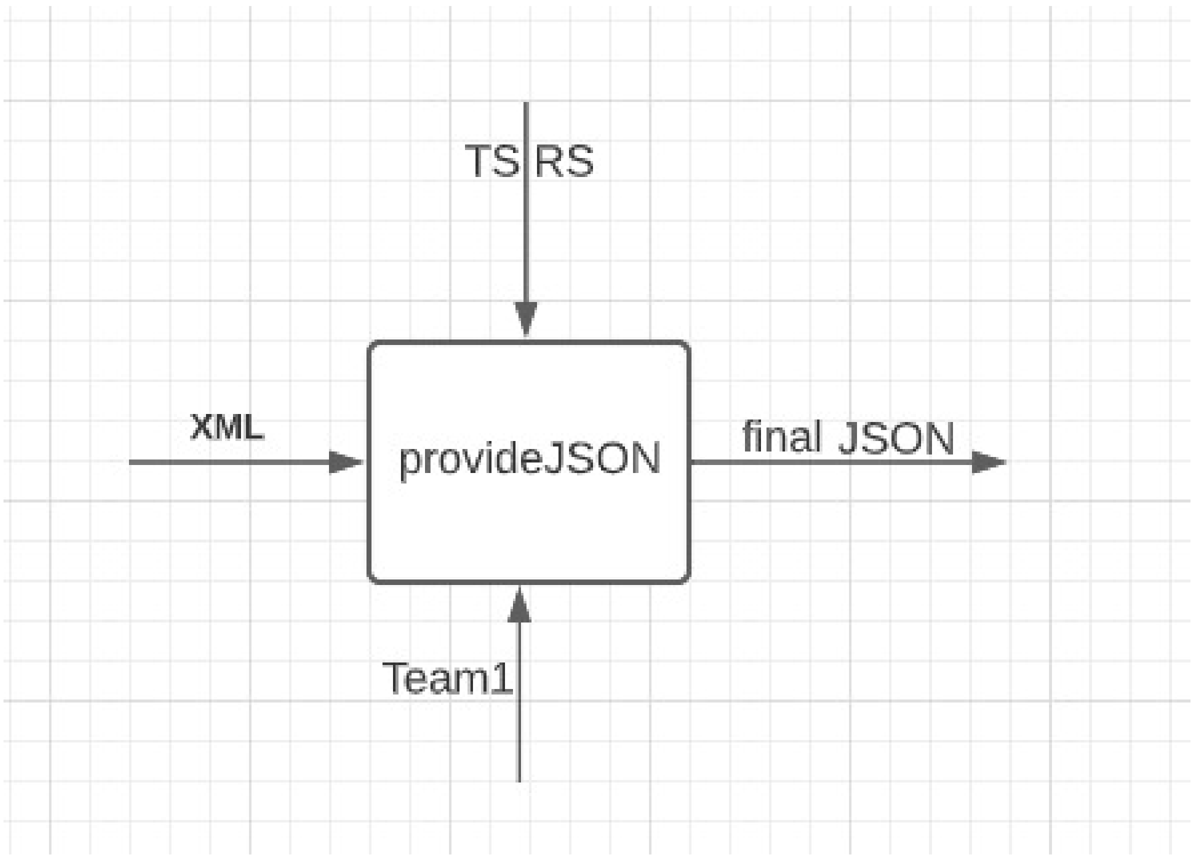
4 Class Diagram



5 Deployment Diagram



6 Context Diagram


TS RS
XML
provideJSON
final JSON
Team1