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

Area's 5 Requirement specification

Done by:

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

Supervisors:
Virgilijus Krinickij,
Gediminas Rimša

Vilnius
2022

Contents

1 Purpose of the team's deliverable

1.1 Intended Audience and Intended Use

The company is our client, and they will obtain a software system that will help company personnel create documents describing how exactly the solar system should be built.

1.2 Purpose

We will create a solution that will tie everything together into a single working software tool that takes a 3D model file along with any user input needed, and then produces the documents that can be sent for approval.

1.3 Possible Risks / Complications

There are a couple of possible risks that might occur during the task:
1. Other teams will have problems with their parts.
2. Troubles communicating with other teams.
3. Implementing everything together.
4. Different data formats are provided by each team.
5. Other teams not finishing their parts in time.

2 A high-level overview of the team's deliverable

We are going to get the data from 4 teams. After that, we will have to create software that will tie everything together into a single working software tool that takes a 3D model file along with any user input needed, and then produces the documents that can be sent for approval.

Our software is going to produce two documents:
1. Overview of the whole site. The purpose of this document is to show the reviewer an overview of the whole system in one place. The following will be displayed in a single image:

a. All roof faces on all buildings.
b. All fire ventilation setbacks and pathways.
c. All solar panels.

2. View of each roof face with solar panels. The purpose of this document is to let the reviewer verify in which wind pressure zones the solar panels will be mounted. For each roof face with solar panels, the following should be displayed:

a. Roof face, with edge type printed next to each edge.
b. All solar panels on this roof face (so it would be visible which solar panels fall into which wind pressure zone).
c. Wind pressure zones.
d. Calculation showing the wind zone width that was used.

3 Complete functional requirements of the team's deliverable

1. Get the information from other teams, by communicating with the team leaders to get the XML document or Java libraries.
2. Parse data, by creating an algorithm.
3. Combine the data into a single XML file.
4. Create the image from parsed data.
5. Create a PDF with images and descriptions.

4 The most important quality attributes (non-functional requirements) of the team's deliverable

1. Security: we are going to check all the data that we receive to make sure that this information is not corrupted and in the right form.
2. Availability: we are going to use Java to create the application that Linux, Windows and macOS can run. Also, our application can be used without the internet.
3. We will add operation reliability and we can check the input cases.
4. We are going to create a simple UI to provide usability.

5 Implementation plan of the team's deliverable

5.1 Communication and preparation plan

1. Divide into groups.
2. Create a group chat.
3. Create a team chat.
4. Run through the whole project to understand the goal and what we need to do to make it.
5. Agreeing on a language (with all teams) that our software will be developed.
6. Communicate with each team about their job, possible results, and all required information to make a final product.
7. Communicate with each team about their progress.

5.2 Documentation and presentation plan

1. Write Requirements specification.
2. Write Technical specification.
3. Write weekly mini-reports.
4. Present final product.

5.3 Development plan

1. Choose how and with what tools we are going to build our part
2. With information from other teams, create data prototypes
3. Create software for each team's data
4. Using separate software, create the final software
5. Test everything with prototype data
6. Get final products from other teams
7. Analyze all the data
8. Make sure the data we got is correct and safe to use
9. Test if the data we got works on our software
10. Make changes if needed