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

Information Technology II year

Requirements 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

Terminology

User	**The engineer that is using the system**
Customer	**The person, with whose roof data the engineers will be working, and who will receive the end result of the whole system**

Team Distribution

Team Leader	**Jomantas Užusinas**
Developer	**Vilius Juknevičius**
Developer	**Dinas Majauskas**
Developer	**Titas Majauskas**
Developer	**Sakalas Stasiulis**

1 Purpose and Overview

1.1 Purpose of The Document

This requirement specification will mainly act as a guideline for Team 3 on the common goal of our team and the idea of the project. It will be available for the Team 3, Team Leaders from other teams and the Project Leader to grasp the understanding of our team‘s work and future goals.

1.2 Purpose of The Team

Our main goal as a team is to create an addition to the main system of our project which will optimize the placement of solar panels on our customer‘s roof by following requirements specified by the manufacturer of said panels.

1.3 High-Level Overview

We, as the developers of this functionality, are creating it with the intention to ease the physical labour of our clients. The functionality will be provided in a form of a high-level overview from team to team:

- The position of the panels will be calculated according to the data received from Team 2 - the panels will not obstruct paths stated in fire code, thus they will not be placed on certain parts of the roof. Besides fire code regulated paths, other non-placeable areas will include: chimneys, skylights, etc.
- The algorithm will also maximize the amount of panels that can be placed on a roof. Made calculations and generated data will be delivered to Team 4 and Team 5 to use according to their requirements.

2 Functional Requirements

2.1 High Priority

- The system, using given data, will automatically calculate the best solar panel distribution on the roof surface leaving optimal gaps between the items.
- The system will identify chimneys, skylights or other obstacles on the roof and will not mark them as solar panel installation friendly parts of the roof.
- The algorithm will try to maximize the amount of solar panels on the roof.

2.2 Low Priority

- Have some user-friendly interface which will help the user of this system to manipulate data and receive certain results.

3 Quality Attributes

- **Availability:**
  - The functionality of our area will be available as a library to use in the main software, which will come in a form of a desktop app.
  - The library will be accessible through Python package installer (pip)
- **Usability:**
  - The functionality created by our team will contain user-friendly, straight-forward approach.
  - The necessary data will be uploaded onto the system prior to the usage of our functionality. Thus, relieving the client of redundant work with files and data.
  - After our client is done with mandatory modifications of said data, it will be automatically available for future processes, functionalities.
- **Compatibility:**
  - Team 3‘s part of the system should be compatible with other teams:
    - Team 2’s data will be compatible with the system.
    - Team 4 can receive processed data from Team 3.
  - Data received from other teams should work with our system and information generated by our algorithms should be reusable by our colleagues.
- **Reliability:**
  - System should work if put under a lot of stress:
    - The system will not stop working when there is >2 users.
    - If >1 file is uploaded in the system, it will still be working.
  - Wrong inputs, corrupted data should stop the system from working, but not make it inoperable entirely.
- **Security:**
  - Users will be working separately, meaning that changes or inputs of separate users working with the same software will not be visible to the other user.
  - Personal information through received data will not be accessible to unauthorized individuals.

4 Future Plans

As it is a understandably complicated project, our team decided that we will distribute our work in said versions:

- **09.05 - 09.09:**
  - Introduction to the project
- **09.12 - 09.16:**
  - Team formation
  - Team leader selection
  - Project area selection
  - Creation of Gitlab repository
- **09.19 - 09.23:**
  - Creation of requirements specification draft
  - Team meeting to understand the project area
  - Move the group repository from Gitlab to Github
- **09.26 - 09-30:**
  - Learning the basics of Python
  - Team repository in group's Github
  - Fix up requirements specification draft
- **10.03 - 10.07:**
  - Creation of separate MS Teams channel for the team
  - Finishing the requirements specification draft
  - Try to move requirements specification draft from MS Word to Overleaf (Latex)
- **10.10 - 10.14:**
  - Learning Python Basics
  - Experimenting with 2D drawing in Python
- **10.17 - 10.21:**
  - Learning to draw UML diagrams
  - Cloned Bpypolyskel library
  - Using demo to draw 3D model
- **10.24 - 10.28:**
  - Writing Technical Specification

  - Solving problems of Bpypolyskel
  - Learning how to rotate a matrix

- **10.31 - 11.04:**
  - Submitting Technical Specification

- **11.07 - 11.11:**
  - Writing algorithm to change coordinates of the roof according to limitations
  - Algorithm to calculate the area of roof without changed coordinates and roof with changed coordinates
  - Algorithm, that calculates the maximum amount of possible solar panels on the roof
  - Drawing the model of the roof with solar panels on it.

- **11.14 - 11.18:**
  - Adding the ability to mark the solar panels on the roof

- **11.21 - 11.25:**
  - Algorithms accurately calculates the placement of the solar panels on the roof

- **11.28 - 12.02:**
  - Algorithm to maximize the solar panels on the roof is implemented

- **12.05 - 12.09:**
  - Last tests of the system before publishing it

Overleaf read only version - *click here*
Github of this documentation - *click here*