Olivia Saukonoja K438991

Miina Rautakorpi H291552

Malla Nyrhinen 50287776

Sini Kivioja H293970

Design document

High level description

The software monitors real-time data on greenhouse gases and compares current data with historical data on greenhouse gases. Real-time data comes from SMEAR API and historical data comes from STATFI API.

We are using C++ language, QtCreator and Qt's libraries for the project, because everybody in our group has experience of it. For plotting and visualizing the data we are using QCustomPlot library. It was downloaded from qcustomplot.com and the needed header and source files were added to the code. QCustomPlot is used in PlotWindow.

The chart 1 will explain the high-level description and show how the classes are linked together. We selected this structure, because it's easy to keep the UI and the data handling separate from each other. Error handling is also clearly its own class.

Boundaries and interfaces

MainWindow starts the user interface application. It creates the smearApi-, statfiApi- and dataHandler-classes after Visualize-button has been pressed. SmearApi and statfiApi fetch the user's selections from MainWindow. SmearApi gets the data from SMEAR's API and statfiApi from STATFI's API with GET-request.

Then dataHandler fetches the data from smearApi and statfiApi and modifies the data. DataHandler creates PlotWindow-class which fetches the modified data from dataHandler and then creates the visualizations. In errorHandler-component we can handle the errors when they appear.

Design solutions

We decided to put the SMEAR and the STATFI decision at the top of the front page, because it affects on what to ask next from the user.

If the user selects SMEAR, he/she will be asked about the time period, monitoring station, greenhouse gas and different ways to present the data. Different monitoring stations will be represented in different graphs, and different greenhouse gases will be represented in different lines inside those graphs.

If the user selects STATFI, he/she will be asked about the time period and dataset. Then the data will be shown in a graph. Different datasets will be represented in different lines.

If the user wants to compare the SMEAR and the STATFI data, he/she can select everything related to the data. Then the SMEAR data and STATFI data will be shown in separate graphs (meaning 2-4 different graphs, depending on how many monitoring stations have been selected).

In case user selects something that leads to some data being unavailable, it is shown as “grey” to indicate that the user cannot select it.

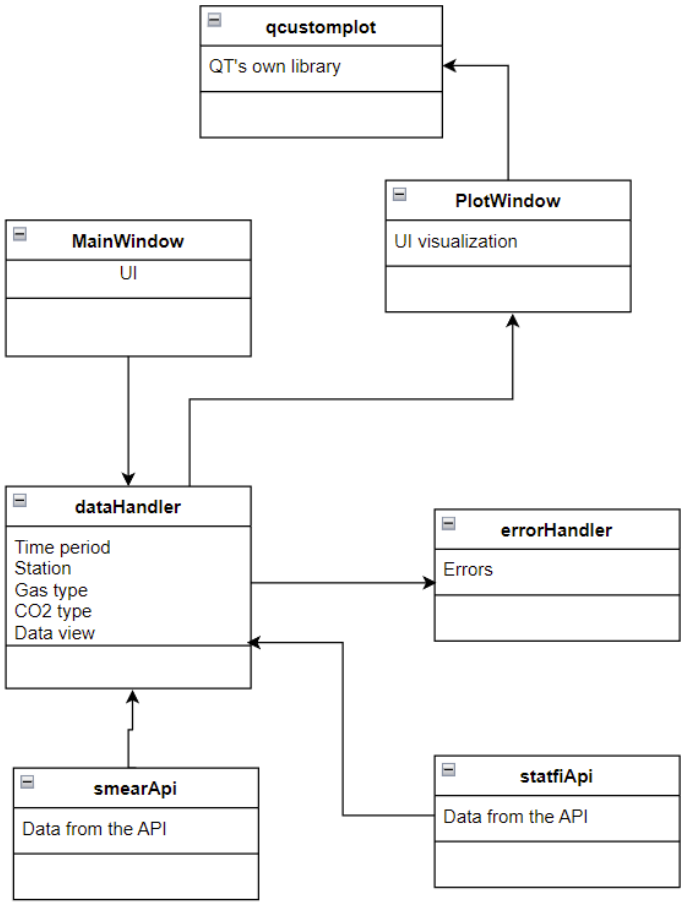

Chart 1

Self evaluation

We have been able to stick to the original design quite well. The UI in the program is really similar to the prototype we made before any coding. The difference is that there are two calendars for both STATFI and SMEAR. Because four different calendars instead of two, the layout changed a bit too, but all the same questions and options are included. The class structure was also able to stick to the original plan, only some names were changed along the way.

Design corresponds to quality so that the classes are clear, and the mistakes are easier to find. Responsibilities of the classes are clear and well defined. The planning and prototype are really similar to the product at the moment, so we think the planning was a success. It was easy to start implementing the program based on prototype.

We think the requirements were taken into account so well at the planning, that there really isn’t many changes we would have to made to be able to implement the rest of the features.