

Unnamed

# Design Document

COMP.SE.110-2020-2021-1 Software Design

Joonas Pelttari  
Kalle-Henrik Raitanen  
Jani Uolamo  
Sipi Ylä-Nojonen  
21.2.2021

## High level description

The applications GUI is implemented with QtQuick using QML. Other classes for application include Model, Controller and DownLoader which interfaces are further discussed in “Boundaries and Interfaces” section.

QtCharts is included as additional Qt-library to easily present data visualizations in a user-friendly way.

Controller class is responsible for processing and delegating user requests from the user interface whereas Model stores downloaded data and updates user view according to orders from controller.

Downloader class is used to handle connections and fetching data from internet sources to be stored by Model.

Storage class is used for storing user preferences regarding automatic datacollection etc.

## Boundaries and Interfaces

### Controller

- fetchData(url), calls for Models interface to initiate data download
- renderData(chartName, seriesName), requests Model for rendition of changed data in view.
- showGraph(chartName, seriesName), requests for view to change visibility of respective graph
- scaleView(chartName), request for View to rescale ChartView timeline

### Model

- fetchData(url), calls for DownLoader to download data
- getPointSeries(name), returns data points respective to series defined by name eg. temperature, power production
- renderData(chartName, seriesName), pushes request for View to update data in UI.
- 

### DownLoader

- load(url), downloads data from url source

### View

- renderData(chartName, seriesName), replaces data in UI with new supplied by Model.
- showGraph(chartName, seriesName), show or hide graphs respective to controller requests
- scaleView(chartName), scale ChartView to different timeline

### Storage

- fetchValue(parameter), get currently saved value for setting
- editValue(parameter, newValue), modify saved setting

## Class diagram

