Design documentation

Description

The program is a general-purpose electricity analytics application. It combines Finnish electricity production and weather data, displaying correlations between different weather metrics and specific types of electricity production methods.

Possible choices for weather types include temperature, cloudiness and wind. Electricity production methods include solar, wind and water. The data can be viewed in graphs that display either a day, a week, or a month of data at once. The chosen parameters can be saved as a preset, for future viewing.

Architecture

The program is implemented as a monolithic Java application with a typical service decomposition of splitting related functionalities into their own classes. Being well-defined, this allows for splitting the application into a microservice architecture relatively easily, should the need for further scalability arise.

Modules

The preliminary plan is for the application to consist of at least the following components:

- Weather fetcher – A class which houses the functionality for fetching data from the FMI API.
- Energy data fetcher – A class which handles fetching data from the Fingrid API.
- Logic service – A class that contains the business level logic for processing the data.
- Utilities – A class for general utility methods.
- Front end – A class containing the JavaFX implementation.

Additionally, other classes might include for example the entities for well-defined data objects. Figure 1 shows an epic diagram depicting the application. Direction of the arrows indicates the direction of the data.

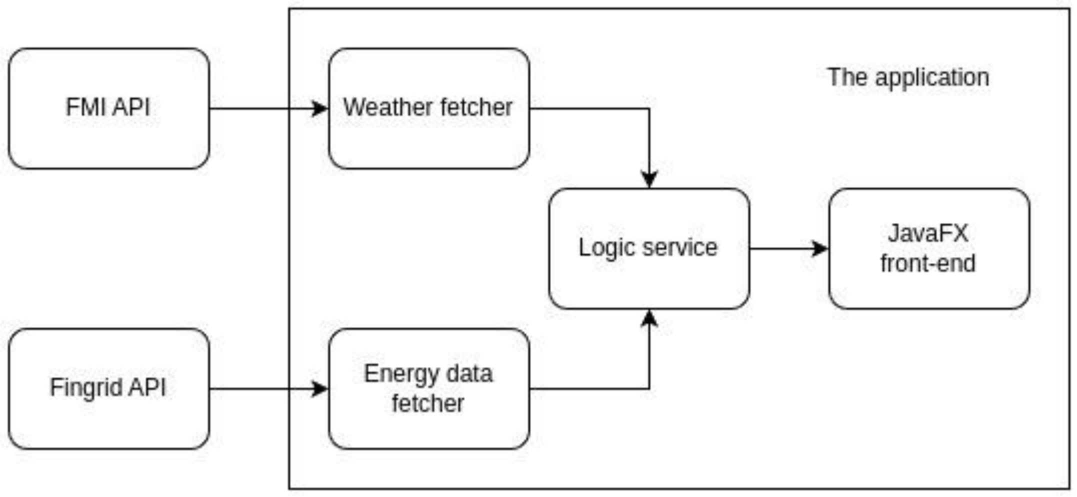

Figure 1: Diagram of the application's basic components.

Implementation

Since this is a desktop application, the UI is implemented using JavaFX. However, the application is designed to allow for a possible REST based web or mobile front-end to be implemented easily. This would simply require a new class for defining the endpoints to handle REST requests.

Fetching data from the APIs is implemented in separate classes for each API. This is to make a clear distinction between different data sources and to simplify the possible addition of new APIs later. Data from the fetchers is combined in the application-level logic service. In addition, the module architecture allows for easily adding a database to the application in the future by creating classes for data access objects.

Since there is no database implementation for now, the current idea is to save any possible data only for the current instance of the application. For example, when saving presets of chosen graph parameters, the application retains the presets only until the application is closed. Since the application is designed in a way which makes integrating a database easy, this can be expanded to actual functionality, should the development continue after the project.

APIs

The software needs electricity consumption data as well as weather data. Finnish electricity data will be fetched from Fingrid's APIs. Fingrid provides APIs for electricity production data which can be filtered by production method. This data updates every 15 or 3 minutes and is calculated as an average value for the 15- or 3-minute period.

FINGRID API

For fetching the weather data an API from Finnish Meteorological Institute will be used. This API provides data for all the relevant weather metrics. Recent data is provided with an accuracy of up to 1 minute.

FMI API manual