Weekly Exercise 3: Implementing a GUI

In this exercise we will practice making graphical user interfaces. The idea is to take the prototype from the first exercise and implement the user interface in practice, with the (object-oriented) language and IDE of your choice. At this point there doesn't need to be any real functionality yet.

Guidance for this exercise is available if you use Qt Creator (C++) or NetBeans (Java).

Links to resources:

- Git course: https://plus.tuni.fi/tie-git/
- Linux remote desktop instructions: https://intra.tuni.fi/en/handbook?page=2640
- Azure virtual desktop instructions: https://intra.tuni.fi/en/handbook?page=30606
  - Note that you can you use Azure virtual desktop in a browser. See section "Using Azure Virtual Desktop in a browser step by step"

Preliminaries

We will continue from the first weekly exercise where you made a prototype for a fitness tracker desktop application. Read the application specification from the first exercise to remind yourself what was included. If you didn't attend the first exercise, you can use the example prototype in the course repository as a reference. Make your implementation so that it can easily be built upon in the future.

Repository link: https://course-gitlab.tuni.fi/comp.se.110-software-design_2023-2024/weekly-exercises/-/tree/main/solutions/ex1

Implementing the user interface

The user interface should have at least the following, implemented how you see fit (bolded elements first and foremost):

- **Some kind of graph view component (no actual graphs yet but a placeholder)**
- **Display selections for device/group, data type, and timeframe with ready options**
- **Some way to manage the user's groups (no functionality yet)**
- Something happens when data is supposed to be displayed (for instance, a placeholder text appears in the graph view when a button is clicked)
- A possibility to upload simple data (text) from a file, the data is then displayed
- A functioning "quit" button that closes the application

You should also think about how you would actually implement displaying data as a graph in your interface, and what kind of classes you would need.

When you're ready, demonstrate your solution to the assistant. Don't forget to put your work in your personal repository to have it available for the next week's exercise.