Final project 2020Z

Course: Python programming and data visualization Lecturer: dr inż. Robert Szmurło

> *The data for this project have been made available courtesy of Mr Paweł Zawadzki, who is registering them for his Phd thesis research.*

As a final project, you should provide a web based application written in Python using the Plotly Dash framework for data visualization. The application should present a graphical visualization of measurements of a device device for monitoring walking habits and patterns of elderly and disabled persons. In Fig 1. you can see a photo of the device. The device was entirely designed by Mr Zawadzki and it based on the Arduino platform. During the project you will a web interface gateway to the simulator of real time measurements. You will be using an HTTP public API for accessing the instantaneous simulated measurements.

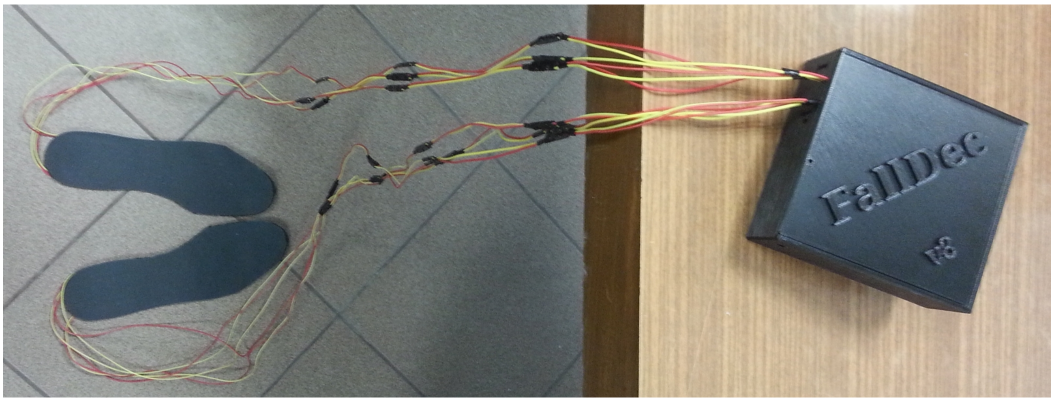

Fig.1 Device setup overview

The monitoring device simulator prepared for this project provides 'real time' measurements of pressure of feet on the ground of real people. Six sensors are located in shoe inserts - three for each foot. With the layout presented in Fig.2. You can use the information for preparing your visualizing application.

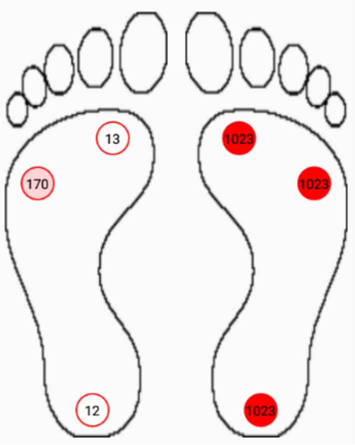

Fig.2 Sensors layout

More details on the device can be found in the attached supplementary material.

For the purpose of the final Python programming and data visualization project an API web service with simulated 'real time' measurements has been made available. Six traces are available with assigned Patient IDs in array the following list: `[1,2,3,4,5,6]`. The service is available only when connected to Faculty of Electrical Engineering vpn (https://vpn.ee.pw.edu.pl) by accessing the following the link:

```
http://tesla.iem.pw.edu.pl:9080/v2/monitor/2
```

The last URL argument is a PatientID from the list. You can test the API with `curl` command:

```
curl http://tesla.iem.pw.edu.pl:9080/v2/monitor/1
curl http://tesla.iem.pw.edu.pl:9080/v2/monitor/2
curl http://tesla.iem.pw.edu.pl:9080/v2/monitor/3
curl http://tesla.iem.pw.edu.pl:9080/v2/monitor/4
curl http://tesla.iem.pw.edu.pl:9080/v2/monitor/5
curl http://tesla.iem.pw.edu.pl:9080/v2/monitor/6
```

The API returns a **single measurement** in a self-documented JSON format. Thus your application to fullfill all requirements specified later in this document will have to store time traces of the sensors itself in some storage to allow viewing history and periods with detected anomaly.

The application should provide an attractive and clearly readable interpretation of 6 traces measuring foot pressure on the ground in three locations for each foot. When designing the visual representation you should take into account the Tuftes rules and 10 rules for better graphical data presentation.

Your task is to design an attractive interface which will allow to analyze the process of walking (animations are welcomed). You can use various technologies for the visualization. In the final report you should include the motivation for your final design of the application. You should describe the expected audience and their expectations. You should describe in the report why you have presented the specific figures and plots and you should explain what information you were trying to present in the plots and figures.

Functional requirements

Your Python web application:

- should present **process of walking** in real time (I suggest some foot visualization with sensors which can change size on the basis of pressure values, name of patient, interactive time traces allowing to analyze the trace, identify anomalies, identify common patterns, highlight patterns, search patterns are welcome);
- should register and store in some storage (file on a disk, sqlite database - your choice) the **history of walking for at least 10 minutes backward** to current time to allow review of historical values;

- should register in a separate database and allow to trace back the parts of a trace which triggered an **anomaly alarm** (the information about the alarm will be made available by the monitoring system and - your task is to just register parts of the trace in your database, label them with time and allow to graphically view; you do not have to implement any signal processing algorithms for anomaly detection ;-)) the purpose of this functionality is to provide a medical staff possibility to find and problems;

Data

The data for the system should be accessed from an on-line available endpoint available at the following address.

```
http://tesla.iem.pw.edu.pl:9080/v2/monitor/2
```

API Documentation

```
http://tesla.iem.pw.edu.pl:9080/v2/static/index.html
```

The id's of sensors are assigned as below:

id	name	meaning
0	L1	left foot front
1	L2	left foot middle
2	L3	left foot back
3	R1	right foot front
4	R2	right foot middle
5	R3	right foot back

Formal requirements:

1. The project should be prepared in two person teams. One person teams are also welcome - but an individual request is required.
2. The project should be submitted in form of a zipped archive ready to be run with one command. For example (this is example!):

```
export FLASK_APP=hello.py
flask run
```

3. The project must contain a short users manual document with description of implemented architecture of the solution. It should not be between three and five A4 pages (no longer than 5 pages). You can use Latex if you wish, but when saying A4 pages I mean pages with content! Latex is very clear to read but it also generates a lot of empty space. (https://www.overleaf.com is suggested, but not required.)
4. You should also submit report summarizing and explaining your design decisions about the visual presentation of the data and the implementation architecture. It

should contain screenshots of your applications and all the parts of the design should be explained and motivated.

Final notes

The project you have to prepare is not very complicated. It is composed of a few graphs and animated image. Keep in mind that the purpose of the project is to create a reliable application with attractive design. This takes some time! I expect that you will explain your design decision (both visual and technical) in writing! The architecture of the project is composed of the backend and frontend. The backend should be implemented with Python, Dash and flask server. You can but not must implement custom Plotly Dash components.