

# Final project for Python programming and Data visualization course

Marharyta Kruk, Krzysztof Dąbrowski

January 7, 2020

## Contents

<b>1</b>	<b>Intro</b>	<b>2</b>
1.1	How to run . . . . .	2
<b>2</b>	<b>Technical details</b>	<b>2</b>
<b>3</b>	<b>Visual design</b>	<b>2</b>

## 1 Intro

Our task was to make web based application written in Python. Our application presents a graphical visualization of a device for monitoring walk habits and patterns of elderly and disabled persons. This device provides real-time measurements of pressure of feet on the ground, so our application's components are real-time based.

### 1.1 How to run

Before starting project make sure to connect to PW VPN to allow access to data API.

Project is containerized with Docker. To start it simply run `docker-compose up` in main directory. After that step open `localhost:8050` in browser.

## 2 Technical details

We use Plotly Dash libraries for our project. We have chosen them because of their elegant simplicity and wide functionality. All elements in our project are parts of the Dash layout, each element represents dash core or html components. We also use built-in Store components to store some data (Current person id and data about last anomaly), so this data is stored in secure way. We store current person id in Session store, so each user can have their own page which will not affect the others.

We have access to the data of 6 persons, so we implemented tabs that changes current person information. As was mentioned above, we store this information in session storage, so current person can be different for different sessions. Current person id manages all other components. We have implemented tabs via dash core components Tabs.

For data storage we have chosen to use Redis database, because of its simplicity. We start acquire data with program beginning. Next, we use data from Redis storage for real-time visualization. For visualizations, we use only 20 last records, because it is optimal amount for fast real-time visualizations, but, of course, we store more.

All our project we have packed in Docker container, so it is easy to build it and run.

## 3 Visual design

We visualize feet information in table. Our table represent Dash core component Graph, which uses component Table from Plotly graph objects. This decision was made because we use dictionaries for building table, and we also store data in Redis database in the form of dictionary. Another plus of using this table is possibility to colour cells depending on their values (as we have implemented). Next, we visualize data via two graphs. First is a graph that shows current

values of sensors on the feet, and also change lines of feet with increasing or decreasing pressure.

Second one is an indicator, which represents each sensor and how value on this sensor changes. We use there the same tabs, co user can easily switch between sensors.

One the end we have histogram of anomalies and label with information about last anomaly, which is visible only when it captures first anomaly. We have decided to use histogram because this graph in a good manner shows density of anomalies.