Taitotalo Oy

Python3Typo

Tanel Levitski

**Projekti: Veren- ja ilmanpaine tietovertailu**

**https://github.com/DEtanel/verepaine**

Helsinki, 2023

1. **Proektin tarkoitus**

Projektin tarkoitus on oppia taitoja, jonka avulla voin omatoimisesti kehittämään Python kielellä erityyppisiä ohjelmistoja. Ohjelmistojen koodauksen jälkeen – olla valmis esittämään opitut taidot sekä ohjelmistot, lisäksi osata käyttämään myös Microsoftin "Visual Studio Code" kehitysympäristöä.

1. **Ohjelman tarkoitus**

Ohjelma, joka tallentaa tietoa ihmisten verenpaineita ja näyttää sen visuaalisesti käyttäjälle. Lopputuloksena voimme visuaalisesti vertailla verenpaineen ja sään vaikutuksia toisiinsa - sekä ymmärtää onko säällä vaikutuksia ihmisen yleiseen hyvinvointiin. Ohjelma kerää myös netissä ilmapainetietoja, joita näytä verenpaineen yhteydessä kuvassa.

1. **Edellytykset ohjelmistoon**

- Henkilön verenpaineen käsin syöttäminen ohjelmistoon.

- Helppokäyttöisyys ja toiminnallisuus

- Tiedon helppolukuisuus

- Verenpaineen ja pulssin visualisointi

1. **Vähimmäisvaatimukset ohjelmiston pyörittämiseen**

- Windows 7 tai uudempi

- Internet- yhteys

- Python ohjelmiston asennus tietokoneen kovalevylle, jossa myös tarvittavat tietopankit, jotta ohjelmistoa voi tarkistella. Latauslinkki: https://www.python.org/downloads/

- Python moduulit:

* PyQt5,
* Matplotlib
* Owm

Asennus tapahtuu helposti komennolla Pip Install [moodulin nimi] tai .bat tiedostolla

Tarvittavat lisäpalikat löytyvät .txt tiedostosta, joka on ladattu GitHub palvelimelle, linkki

<https://github.com/DEtanel/verepaine/blob/main/requirements.txt>

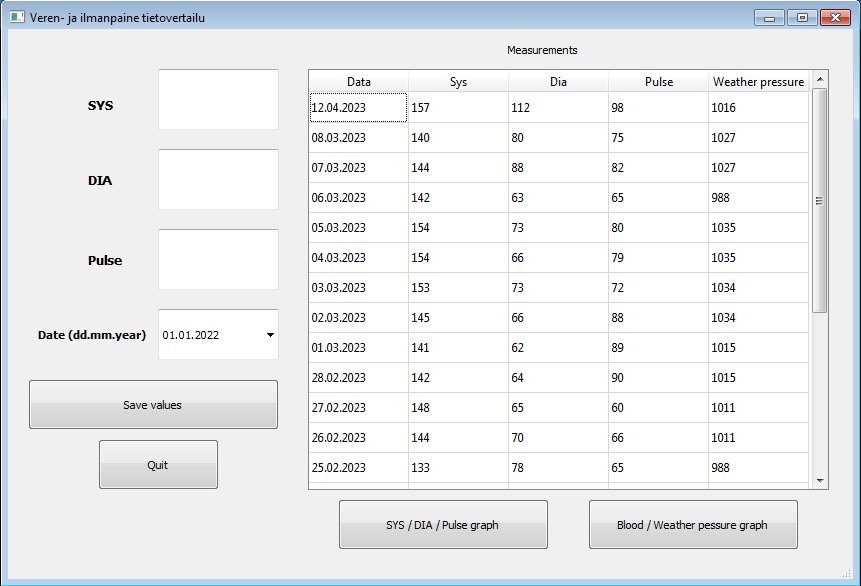
Ohjelmiston lähdekoodi ja testi verepainen/ilmapaine- tiedostot löytyy myös samasta paikasta.

1. **Käyttöohje**

Ohjelman käynnistys tapahtuu syöttämällä Windowsin komento-ikkunaa (*cmd*): *python main.py* kansiosta , mihin on tallenettu kaikki kaynnistyksen tarvittavat tiedot (*main.py, mainwindow5.py*) ja testaus-tiedot \*.txt –muodossa.

Tämän jälkeen aja ohjelma.

Käynnistäessä ohjelman ilmestyy ruudulle graafinen ruutu, joka on jaettu kahteen osa-alueeseen.



Vasemmalla puolella löytyy kohdat, johon voi syöttää SYS, DIA ja Pulse tietoja. Nämä liittyvät standardiin, jolla ihmisen verenpainetta voi mitata. Alla näkyy itse data-päivämäära, joka päivittyy päivittäin automaattisesti.

Oikealla puolella on kaaviokuva, jossa on viimeisen tallennetuilla tiedoilla. Tiedot ovat järjestetty pvm., mukaan uusimmasta vanhempaan. Tämä on tehty tarkoituksella, jotta käyttäjäkokemus olisi mahdollisimman helppokäyttöinen. Haluan, että ylimmillään on uusin tieto mahdollinen, muuten käyttäjien olisi pitänyt selata kaaviota ja etsiä tarvittava pvm., joka tekee ohjelmasta vaikeampaa käyttää.

Lisätessä uusimmat tiedot ja painaessa ”*Save Data*” nappia, syötetty tieto päivittyy järjestelmään, mukaan lukien kyseisen päivän ilmanpaine, ja tallentuu .txt tiedostoon ja kaavio päivittyy.

Painaessa nappia ”*Blood / Weather pressure graph*” uuteen ikkunaan uusi kaavioikkuna ilmestyy. Ikkunassa on erilaisilla väreillä väritetyt viivat, jotka edustavat SYS- tietoja, eli henkilön verenpaine sekä ilmaston ilmanpaine. Täten henkilö käyttäessä ohjelmistoa voi visualisoida nämä kaksi datalähdettä kaaviossa, ja tarkistella onko tiedoilla yhtenäisyyksiä. **Mikäli viivat ovat suurimmaksi osaksi rinnakkain toisiaan, taikka kaaviossa näkyy yllättäviä poikkeamia – voi tämän perusteella päätellä, että säällä on vaikutusta henkilön yleiseen hyvinvointiin. Mikäli viivat ovat täysiä vastakohtia, eikä näytä samanlaisilta – voi päätellä, että säällä ei ole vaikutusta henkilön yleiseen hyvinvointiin.**

Nappi ”*Quit*” mahdollistaa ohjelmiston käytön lopettamista ja täten sulkeutuu.

1. **Tärkeät havainnot ja rajoitukset**

Jotta datasta saa vertailukelpoista on tärkeätä ymmärtää, että ohjelmisto vaatii käyttäjältä joka päivästä tietoa hänen verenpaineestansa sekä säästä (tässä tapauksesta - ilmanpaineesta).

Tämä tieto on syötettävä vähintään kerran päivässä ja vähintään 7 päivän ajanjaksolla. Ilman näitä tietoja ko., ajanjaksolla on mahdotonta suorittaa analyysia, eikä data olisi realistista tai hyödyllistä. Lisäksi mikäli dataa olisi saatavilla vain 2, 3 tai 5 päivän edestä , sitten pysty suorittaa analyysia siitä, että onko säällä merkitystä henkilön verenpaineesta. Lisäksi on muitakin merkittäviä tekijöitä, jotka voivat sekoittaa analyysia, ei siis pelkästään ilmanpaine.

Tässä vaiheessa on myös hyvä mainita, että ohjelmistossa ei ole historiallista dataa Suomen säästä taikka ilmanpaineesta, mutta on suunnitteilla ja tulevaisuudessa osana ohjelmistoa. Datapankit, jotka mahdollisesti voivat toimittaa ko., informaatiota vaativat jonkinlaista maksua, joka voi olla myös järkevä investointi, mutta vaatii vielä harkintaa. Suomalainen nettisivu [avoindata.fi](http://avoin.fi/), tarjoaa tiettyjä tietoja ilmaiseksi. Vaikka sivulta löytyisi tarvittavaa tietoa, niin on vielä tutkittava API mahdollisuuksia, jotta ko., datan saisi sovellukseen onnistuneesti.

1. **Ohjelmiston rakennus**

Itse ohjelmisto on rakennettu käyttäen moduuli-rakenteista periaatetta, jolloin pääohjelma main.py toimii pohjana, GUI (**G**raphics **U**ser **I**nerface) elementti on rakennettu PyQT5 perustella, joka on aluksi suunniteltu ohjelmassa QTDesigner, joka sisältyy tähän PyQt5-pakettiin nimellä ”tools” ja koodattu mainwindow5.py. Lähdekoodissa löytyy kaikki tarvittavat moduulit ja elementit (mm., ikkunat, painikkeet, paikkat kaaviolle, jne.), joka varmistaa, että ohjelmistoa käynnistäessä kaikki moduulit aukeavat ja toimivat suunnitellusti. Ohjelmisto on myös suunniteltu siten, että lähdekoodia on mahdollista muokata siten, että tiettyjä moduuleja voi irrottaa ja koodata eri-osiin. Käynnistyessä lataukset ja tuonti mahdollistavat oikeiden moduulien lataamisen.

Tämän jälkeen käynnistyy alustaminen, joka päättelee mitkä moduulit käynnistyvät tuonti- vaiheessa. Jonka jälkeen käynnistyy funktiot, jotka mahdollistavat datan prosessoinnin sekä funktiot, jotka liittyvät graafiseen järjestelmään (napit jne.).

Ohjelmistossa on myös seuraavat funktiot:

* *Read\_files()* - funktio avaa tiedot, jossa säilytetään verenpaine ja pulssi mittaukset, jotka näkyvät tarkoitetussa kohdassa ohjelmiston visuaalisessa kaaviossa. Uusimmat tiedot säilytetään tiedoston viimeisellä rivillä.
* *Save\_to\_reversed\_files()* - funktio luomiselle, avaamiselle ja lisäykselle – samalla tavalla kuin yllä mainittu funktio, mutta säilytys on toistenpäin, eli uusimmat tiedot ovat ensimmäisellä rivillä.
* *Funktio*, joka on yleinen tuotu moduuli *maplotlib*:ssa, jonka avulla saadaan graafisia ikkunoita, on visualisoitu käyttäen tietoja.
* Ilmanpaine tietoja on tuotu palvelusta *openweathermap.com*, moduulin nimi on *owm*. Jotta tietoja voi käyttää on tehtävä palveluun käyttäjätunnus ja tämän jälkeen palvelusta saa API-avaimen (api key). Avain oli lisätty lähdekoodiin – tiedot tallentuvat *weather\_pressure.txt* tiedostoon.

 Tietojen säilytys CSV-tiedostossa näytti optimaaliselta, varsinkin kun vertailee, että tietoja säilytettäisiin 5 eri .txt tiedostossa – mutta prosessissa tuli mutkia vastaan ja huomasin, että uusien tietojen lisääminen jokaisen txt-tiedostoon ( *sys.txt, dia.txt, pulse.txt, data.txt. weather\_pressure.txt*) oli minulle helpompaa. Mahdollisesti tulevaisuudessa pystyn optimoimaan lähdekoodia, jotta tallennus onnistui yhdelle .csv tiedostolle.

1. **Testaus**

Ohjelman koekäyttäminen tapahtui Microsoft Visual Studio Code kehitysympäristössä. Testasin jokaista moduulia erikseen, ja virheet korjasin sitä mukaan, kun niitä tuli vastaan käynnistäessä mainittuja moduuleja. Monet virheet olivat syntaksin käskyn tai parametrin virheitä, ja pääsin korjaamaan ne nopeasti.

Testaukseen osallistui muitakin ihmisiä, joilta keräsin arvokasta palautetta. Heidän verepainetiedot kerättiin kirjallisesti, ja testauksessa todettiin, että monet näkivät ohjelmalle paljon käyttöä – sillä testaajista useat halusivat tietää, että onko säällä vaikutusta heidän hetkittäiseen huonoon yleiseen hyvinvointiin, vaiko eikä säällä ole laisinkaan merkitystä.

Elementtien sijainnit sekä koko useasti vaihtuivat kehitystyön aikana – erilaisia vihjeitä ja palautteita on otettu huomioon ryhmän muilta osallistujilta. Lisäksi otin vahvasti palautetta vastaan ohjelmiston testivaiheessa, mm., helppokäyttöisyydessä ja tulevaisuuden mahdollisuuksissa.

1. **Aikataulu**

Projektin aikana huomasin, että useasti käytin allokoitua aikaa muihin osa-alueisiin, jotka vaativat minulta aikaa perehtyä uusiin projektin osa-alueisiin mm., moduuleihin, jotka mahdollistavat ohjelmiston toimivuuden. Lisäksi aikaa meni graafisen käyttöliittymän moduulin oppimiseen (mathplotlib, PyQt5) ja käyttöön. Ongelmaksi koitui, että moduuli oli monimutkainen ja vaati minua käymään monta vaihdetta läpi. Mikäli törmäsin ongelmiin, niin yleensä saatavilla ollut tieto oli jo vanhentunut, eikä auttanut ongelman ratkomiseen. Lisäksi muiden koittamat ongelmat olivat useasti jo vanhentuneelle versiolla. Uusimmassa versiossa komennot olivat erilaiset, taikka käynnistyivät eri parametreilla. Tämä kaikki vaikeutti etenemistä.

Lopputulos:

Itse opiskelu /Web-opetus 2 viikkoa

Ohjelmistokehitys 1 viikko

Ohjelmointi 3 viikoa

Testaus ja optimointi 3 viikoa