

Talousdatan visualisointiohjelma

Ohjelmoinnin peruskurssi Y2, oma projekti: yleissuunnitelma

Markus Säynevirta, 594684, ELEC EST, 3. vsk

6.3.2020

Yleiskuvaus

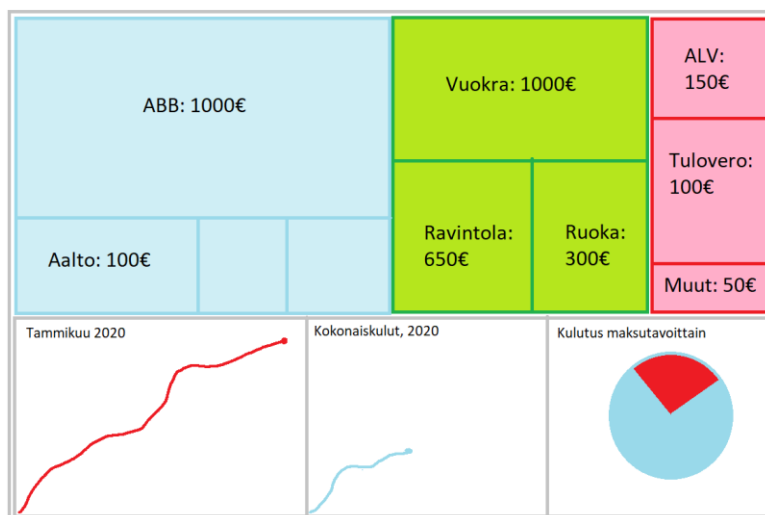
Projektin tavoitteena on toteuttaa json-muotoisen talousdatan visualisointiin tarkoitettu ohjelma, jolla voisi tuottaa erilaista tilastotietoa käyttäjän kulutustottumuksiin liittyen. Ajatuksena on toteuttaa tietyiltä osin S-kanava, Nordea Wallet ja OP:n talouden tasapaino -tilastosovellusten sisältöä vastaava ohjelma, jossa voisi piirtää esimerkiksi blokkidiagrammin tulojen, menojen ja verojen kategorisesta jakaumasta (toteutuksen pohjana esim. Suomen valtion budjettia visualisoiva [tutkibudjettia.fi](https://www.tutkimus.fi/tutkimus/budjetti)).

Raakadatan käsittelyyn on toteutettu myös syksyn C++ -kurssilla (ELEC-A7151) komentoriviohjelma, jolla json-muodossa olevaan tiedostoon voi tuoda automaattisesti pankkien tekstitiedostoina olevia tiliotteita. Projekti jatkaa aiemman C++ -projektin azure-ympäristön käyttöä projektitavoitteiden hallinnassa.

Projektin tavoitteena on rakentaa PyQt -kirjastoon pohjautuva graafinen käyttöliittymä, jonka pääominaisuudet liittyvät datasta koostettujen kuvaajien esittämiseen. Käyttöliittymää tukemaan tarvitaan myös json-datasettiä käsittelevät luokat, joilla datasta voidaan muodostaa visualisointipuolen vaatimusten mukaista syötedataa.

Projekti tähtää vaativan tason vaatimukset täyttävään toteutukseen.

Käyttöliittymän luonnos



Käyttöliittymä aukeaa ohjelman käynnistyessä dashboard-näkymään, jossa on neljä erilaista kuvaajaa. Ikkunan pääosan käyttää [tutkibudjettia.fi](https://www.tutkimus.fi/tutkimus/budjetti) blokkidiagrammi, joka kuvaa koko vuoden tulojen, menojen ja verojen kategorisesta jakaumasta. Ikkunan alaosaan tulee kolme palkkia, joista kaksi ensimmäistä ovat kuluksen kuukauden ja vuoden kumulatiiviset menot ja tulot viivagraafina. Kolmantena on menneen vuoden kulutuksen piirakkakuvaaja maksutapojen mukaan.

Dashboard-näkymän kuvaajista on tavoitteena saada interaktiivisia, jolloin niiden osia voisi tarkastella tarkemmin kuvaajaa klikkaamalla. Osien tarkempi tarkastelu voidaan vaihtoehtoisesti toteuttaa painikkeilla ja valikoilla, mikäli itse kuvaajan klikkaamisen toteuttaminen osoittautuu ongelmalliseksi.

Tiedostot ja tiedostoformaatit

Ohjelman toteutuksen pohjana toimii vuosien 2019 ja 2020 kulutus- ja tulotiedoista json-muotoon koostetut tilastotiedot, joiden aineisto on kerätty pankkien tiliotteiden ja talteen otettujen ostokuittien pohjalta. Ohjelma ei tee muutoksia käyttämäänsä json-tietokantaan, vaan ainoastaan lukee sitä. Pää tietokannan toteutusta esitteleviä mallitiedostoja on repositorion doc-haarassa sijaitsevassa `example_dataset` -kansiossa.

Kuvaajien konfiguraation määrittämiseen käytetään json-tiedostoja, joissa määritellään esimerkiksi, mihin datan tietyt kategoriat sisällytetään kuvaajassa (esim. lounaat, illalliset ja pikaruoka yhden ravintola-pääkategorian alle).

Järjestelmätestaussuunnitelma

Projektin toteuttamiseen käytettävissä oleva aika on varsin rajallinen ja yksilöprojektina testaus syö kehitysaikaa toiminnallisen ohjelmakoodin kehityksestä. Nämä rajoittavat tekijät huomioiden testauksen painopiste tulee olemaan ohjelman sisältämien metodien yksikkötestauksessa, missä tavoitteena on täysi kattavuus. Ohjelmaa voidaan tietyiltä osin myös integraatioiteta. Ohjelman kehityksen lähtökohtana on testipohjainen ohjelmistokehitys, millä voidaan välttyä merkittäviltä testiaukoilta.

Yksikkötestauksessa on tarkoitus käyttää pääasiassa unittest-kirjastoa, ja testejä on pääasiassa tarkoitus tuottaa dataa käsittelevillä apumetodeille, sillä graafisen käyttöliittymän testaaminen automaattisesti tai manuaalitestin olisi todennäköisesti liian aikaintensiivistä.

Yksikkötestauksen metodeilla varmistetaan ohjelmakoodin metodikohtainen toiminta syöttämällä koodille erilaisia variaatioita syötedatasta. Talousdatalaskennan kohdalla metodien syötedatana ovat pää tietokannan variaatiot ja erilaiset konfiguraatitiedostot, joiden yhdistelmä tuottaa erilaisia ulostulevia datasettejä. Testimetodit ovat toteutukseltaan hyvin saman tyyppisiä eri visualisointikomponenttien välillä, joskin niille annettava syötedata vaihtelee ainakin konfiguraatitiedostojen osalta.

Yksikkötestejä ajamista automaattisesti repositorioon tehdyille commiteille, mikäli Aallon tarjoama kehitysympäristö tukee ominaisuutta.