

Proiect IDP

Aplicatie de evidenta a cheltuielilor

1 Descrierea aplicatiei

In ziua de astazi, cu totii realizam un numar ridicat de cheltuieli in fiecare zi. Datorita timpului redus care ne ramane, nu mai acordam atata atentie aspectelor cum ar fi cat de multi bani am consumat in general, relativ la perioade precedente, pe ce categorii de produse am consumat cei mai multi bani etc. Scopul acestei aplicatii este de a satisface acest neajuns, prin implementarea un mecanism de evidenta/monitorizare a cheltuielilor realizate de un numar de persoane (clientii aplicatiei).

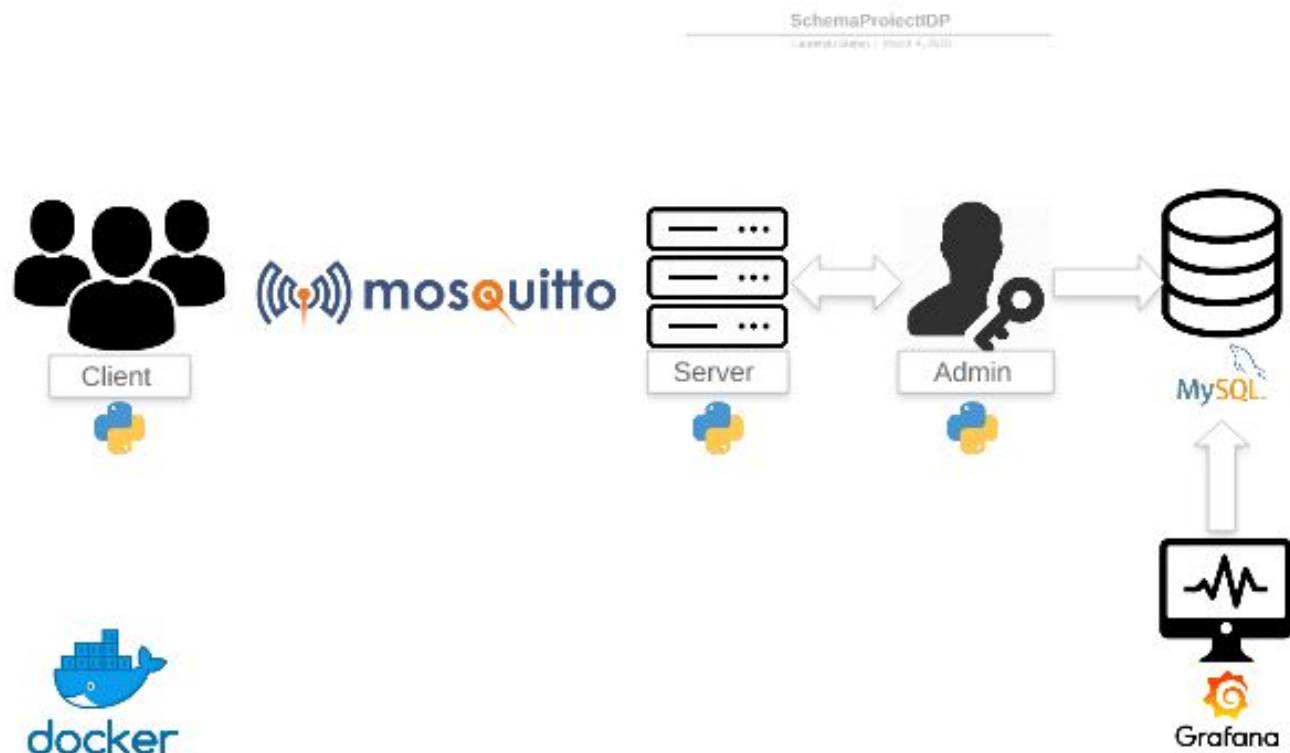
Modul de functionare al acestei aplicatii este unul simplu: fiecare utilizator care doreste sa isi adauge cheltuielile pentru o anumita perioada se conecteaza la serviciul de client; in acesta, printr-o interfata in linie de comanda, se introduc detalii despre ultimele cheltuieli; mai departe, acestea sunt transmise catre un server care insereaza in baza de date aceste informatii ca date noi; tot timpul, ruleaza o aplicatie de monitorizare care realizeaza o serie de grafice pe baza datelor existente in baza de date. Utilizatorul se poate conecta la aceasta aplicatie de monitorizare pentru a vizualiza graficele care il intereseaza pe el.

O asemenea aplicatie aduce, pentru orice utilizator, un numar considerabil de beneficii:

- Graficele realizate pot fi foarte ajutatoare pentru a face diferite observatii: pe ce se produs sau categorie de produse se cheltuie banii cei mai multi, in ce perioade din timpul anului/lunii se cheltuie cei mai multi bani etc. sau pentru a analiza retrospectiv evenimentele din trecut
- Faptul ca se retine intr-o maniera persistenta cheltuielile realizate constituie un avantaj pe termen lung si nu numai

Dezavantaj principal al aplicatiei vine de la necesitatea de a introduce cheltuielile realizate intr-un mod manual, de fiecare data cand utilizatorul considera ca acest lucru trebuie realizat. Totusi, acest neajuns poate fi rezolvat in teorie (cel putin pentru platile online) prin conectarea telefonului sau a aplicatiei de banking la serviciile aplicatiei. De asemenea, un alt dezavantaj o sa il constituie lipsa izolarii (cel putin la inceput) pentru fiecare utilizator. Practic, oricine va putea folosi serviciul de client si sa introduca cheltuieli noi, pe numele acestuia si sa vada in aplicatia de vizualizare grafice pentru orice persoana care a introdus cheltuieli. Se considera totusi ca aplicatia o sa fie folosita de pe acelasi calculator (fiind locala) deci nu ar trebui neaparat sa reprezinte o mare problema.

2 Schema serviciilor



3 Descrierea serviciilor

3.1 Serviciul de client

Serviciul de client o sa fie reprezentat de o aplicatie scrisa in **Python** ce primeste comenzi in linie de comanda de la utilizator pentru a gestiona cheltuielile acestuia. Acest serviciu comunica printr-un broker de MQTT cu serviciul de server, care se ocupa mai departe sa comunice cu baza de date.

3.2 Broker-ul de mesaje (MQTT Broker)

Pentru acest serviciu se va folosi imaginea de Docker **eclipse-mosquitto**. Rolul acestui serviciu este de a implementa protocolul de comunicatie intre serviciul de client si cel de server.

3.3 Serviciul de server

Serviciul de server o sa fie reprezentat de o aplicatie scrisa in **Python** ce primeste comenzile de gestiune de la serviciul client si opereaza cu baza de date prin intermediul serviciului de administrator care are acces direct la baza de date. Aceste servicii vor comunica prin intermediul unui socket.

3.4 Serviciul de administrator

Serviciul de administrator o sa fie reprezentat de o aplicatie scrisa in **Python** ce primeste comenzile de operare asupra bazei de date de la serviciul de server si va opera in mod direct cu baza de date printr-un conector.

3.5 Serviciul de persistenta

Pentru serviciul de persistenta se va folosi imaginea de Docker **mysql** pentru baza de date MySQL.

3.6 Serviciul de monitorizare

Ca serviciu de monitorizare se va folosi imaginea de Docker **grafana** pentru aplicatia de monitorizare Grafana. Aceasta are suport si pentru baza de date MySQL si va realiza graficele aferente in functie de datele introduse in tabelele din baza de date.

4 Comenzi build etapa 1

- Pornire mqtt broker

```
$ docker run -it -p 1883:1883 --name mqtt-broker eclipse-mosquitto
```

- Build client image & start client container:

```
$ docker build -t clientidp .
```

```
$ docker run -it --link mqtt-broker clientidp
```

Repository GitHub:

<https://github.com/laurentiustefan97/ProiectIDP>

Repository Docker Hub:

<https://hub.docker.com/repository/docker/laurentiustefan97/idp>