Katarzyna Szulc
Błażej Szulc

Dokumentacja

repozytorium: https://github.com/rockmind/LoveAndMarriage

Celem projektu jest zbieranie oraz prezentacja danych dotyczących zachorowań na COVID-19. System pobiera tekstowe raporty zarażeń na koronawirusa ze strony rządowej. Magazynuje je w bazie danych oraz udostępnia wizualizacje poprzez przeglądarkę.

Analiza wymagań

Wymagania funkcjonalne

- dostęp nie wymagający logowania
- graficzna wizualizacja danych
    - na osi czasowej
    - na mapie w podziale na województwa
    - na mapie w podziale na powiaty
- możliwość wyboru daty prezentowanych danych
- możliwość wyboru typu danych: liczba przypadków, liczba zgonów, liczba ozdrowieńców

Wymagania niefunkcjonalne

- stabilność
- architektura mikroserwisowa
- konteneryzacja
- użytkownik serwisowy używany do sterowania aplikacją
- codziennie zaktualizowane dane

Opis przeprowadzonych testów

Na system nałożony jest monitoring Grafana + Prometeusz.
Kontrola skalowalności - skrypt pythonowy.
Skrypty testujące poprawność pobranych z sieci i zapisanych danych w bazie danych.
Skrypty do testowania auto-skalowalności aplikacji.
Ponadto system testowany manualnie.

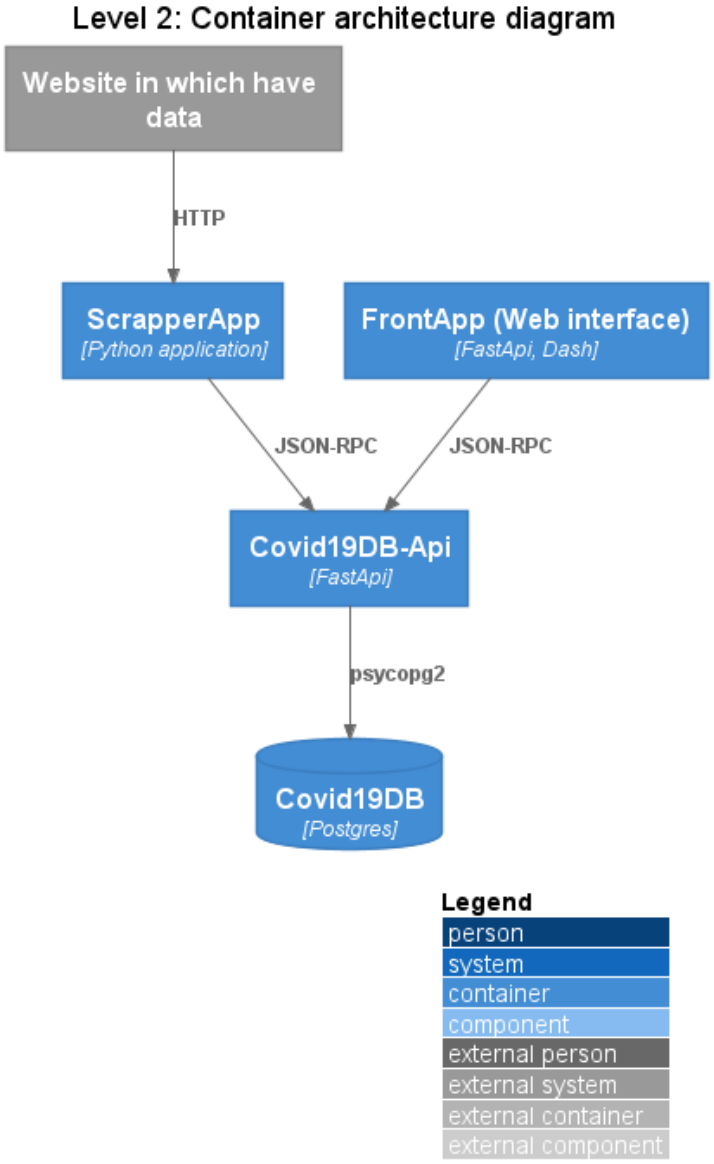
Instrukcja użytkownika

Znając adres udostępnionego interfejsu graficznego można uruchomić aplikację w przeglądarce. Użytkownik ma możliwość interakcji z wykresami:

- wykres dziennej liczby przypadków:
  możliwość wyboru prezentowanego typu danych za pomocą rozwijanej listy, do wyboru:
    - dzienna liczba przypadków
    - dzienna liczba zgonów
    - dzienna liczba ozdrowieńców

  Możliwość powiększania wykresu poprzez zaznaczenie zakresu dat na osi
- wykres - mapa, w podziale na województwa z kolorową skalą. Po najechaniu kursorem myszy na dany region wyświetlają się szczegółowe informacje. Wybór daty prezentowanych danych za pomocą rozwijanego kalendarza.
- wykres - mapa, w podziale na powiaty z kolorową skalą. Po najechaniu kursorem myszy na dany region wyświetlają się szczegółowe informacje. Wybór daty prezentowanych danych za pomocą rozwijanego kalendarza.

Technologie



Wykorzystane technologie:

- **Python 3.10** - główny język programowania wykorzystany w projekcie,
- **FastAPI** - python-owy framework serwerowy, chwalący się jedną z najlepszych wydajności wśród serwerów pythonowych.
- **Plotly/Dash** - framework do prezentacji danych, które pozwala na bardzo przystępne prezentowanie różnego rodzaju danych,
- **postgreSQL** - serwer bazodanowy,
- **SQLAlchemy** - python-owy framework służący do szeroko rozumianego łączenia się z relacyjnymi bazami danych m.in. z postgreSQL
- **Docker** - narzędzie do wirtualizacji i konteneryzacji aplikacji,
- **Kubernetes** (**minikube**) - do opisu architektury systemu oraz minikube do realizacji tej architektury,
- **Graphana, Prometheus** - narzędzie do monitorowania poprawnego działania systemy.