AirQuality

Twórcy: Dominika Boguszewska, Karol Kuc, Krzysztof Sokół, Aleksandra Szymańska

Opiekun projektu: Łukasz Skonieczny

Nasza misja

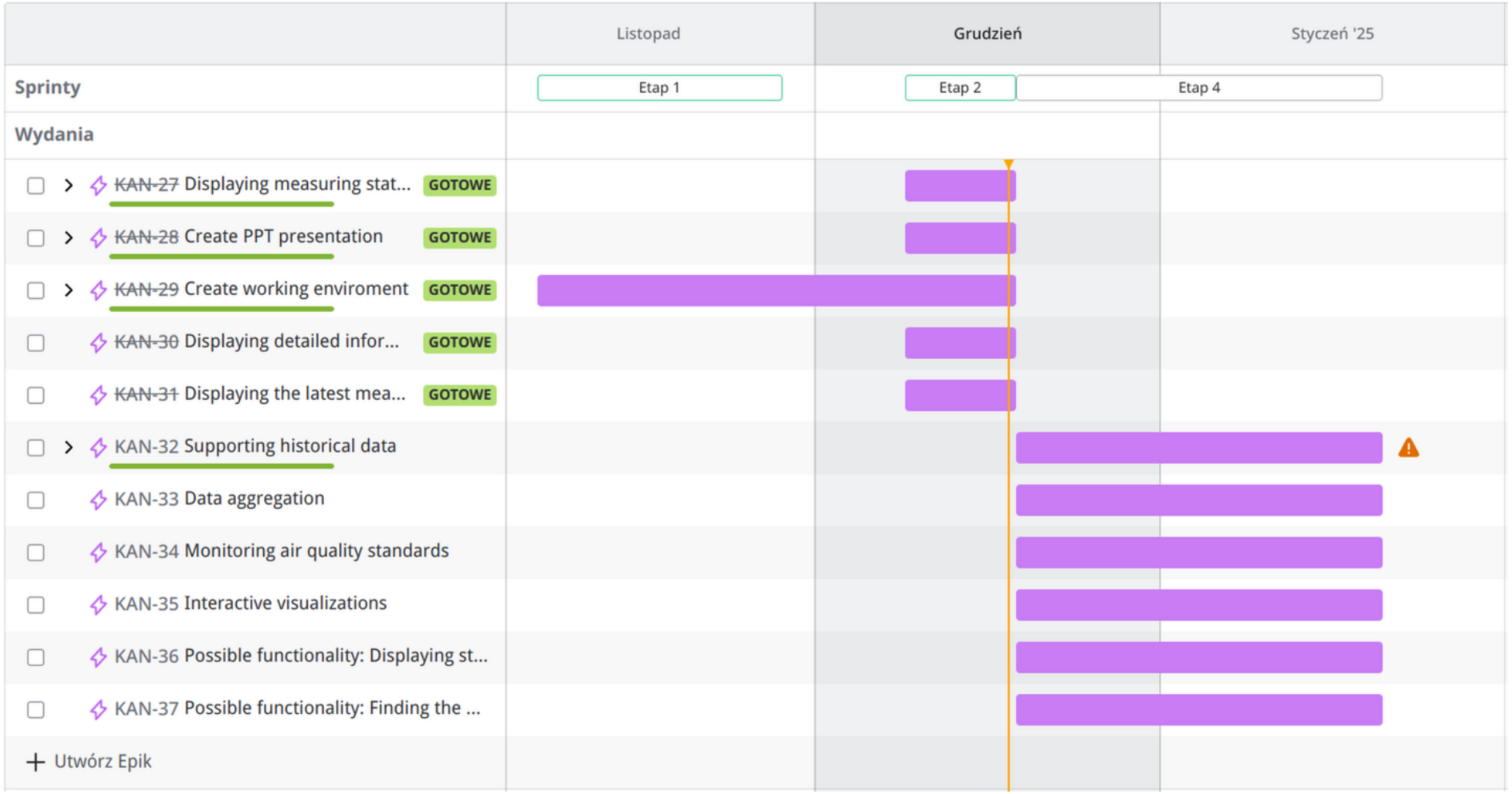
Celem projektu jest stworzenie systemu informatycznego do przechowywania, przetwarzania i wizualizacji danych o jakości powietrza na podstawie publicznie dostępnego API (https://powietrze.gios.gov.pl/pjp/content/api).

System ma wspierać analizę danych historycznych, ich agregację oraz umożliwiać interaktywne wizualizacje w przystępnej formie dla użytkowników.

Posortowane przypadki użycia

- Obecne
  - Użytkownik wyświetla i filtruje listę stacji
  - Użytkownik wyświetla najnowszy pomiar z każdego sensora na stacji
  - Użytkownik wyświetla wykres najnowszych pomiarów
- W przygotowaniu
  - Użytkownik wyświetla informacje przekroczonych normach powietrza
  - Użytkownik wyświetla dane historyczne i wybiera ich ramy czasowe
  - Użytkownik wyświetla dane zagregowane dla różnych poziomów administracyjnych i przedziałów czasowych

Lista funkcjonalności oraz zadań podzielona na 2 iteracje w Jirze



Podział zadań

Projekt realizujemy w dwóch iteracjach: wstępnej(obecnej) i końcowej – tak jak to było przedstawione przy funkcjonalnościach

Dotychczasowy podział pracy:

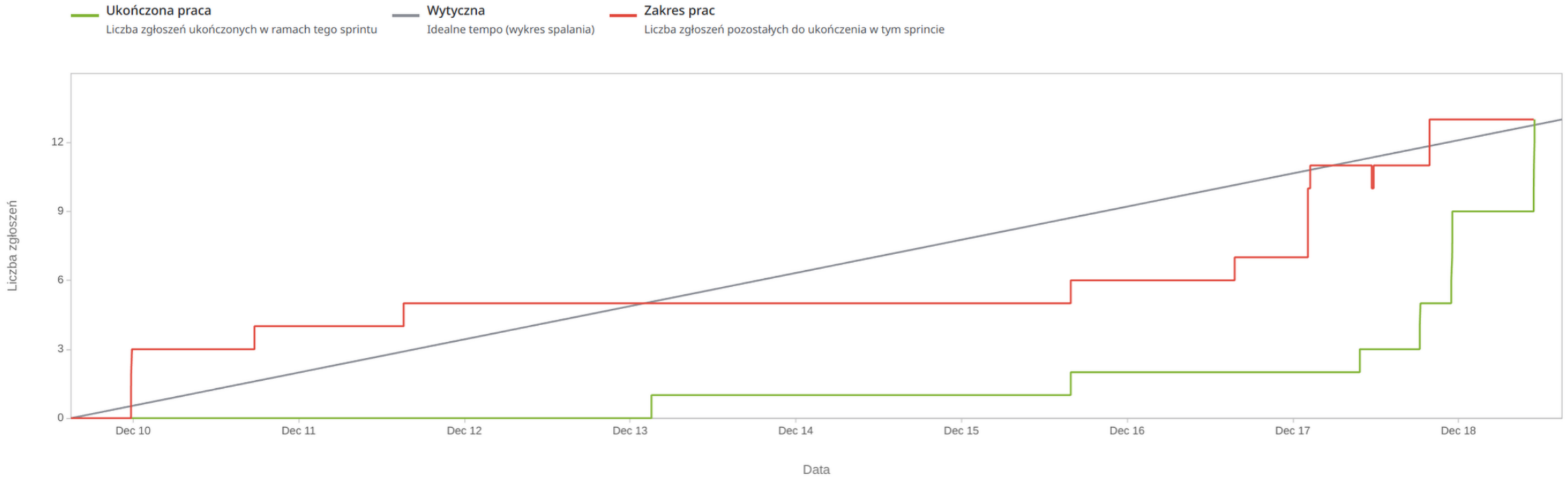
Dominika Boguszewska: Jira, baza danych

Karol Kuc: Backend w Django

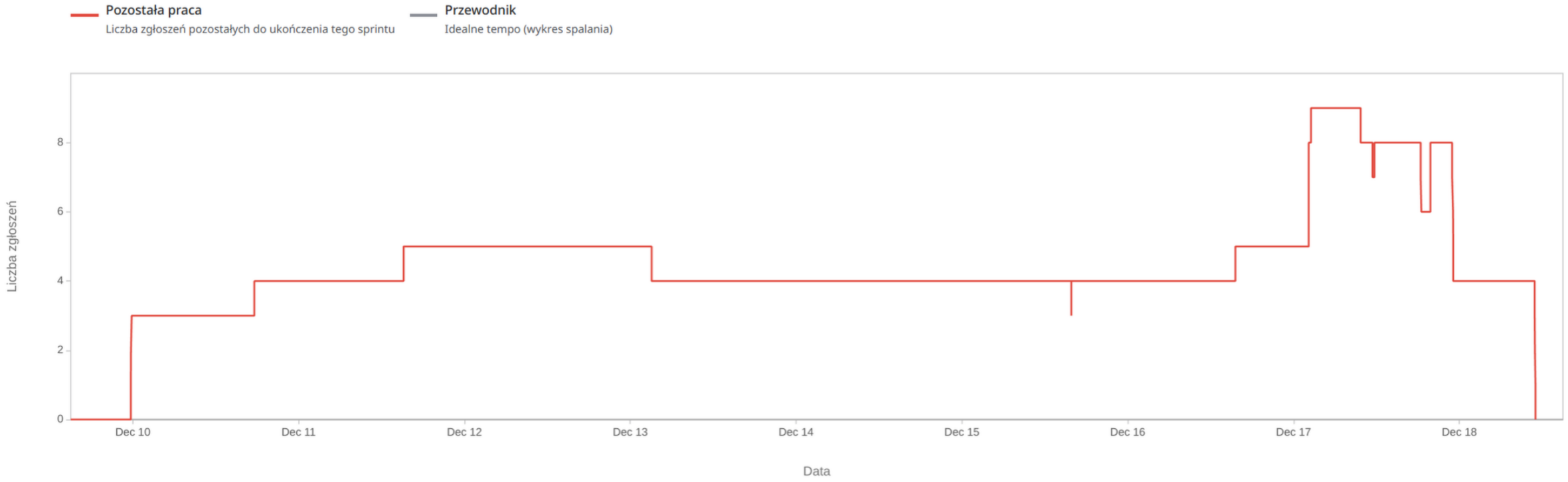
Krzysztof Sokół: Jenkins, deployment

Aleksandra Szymańska: Frontend w Streamlit

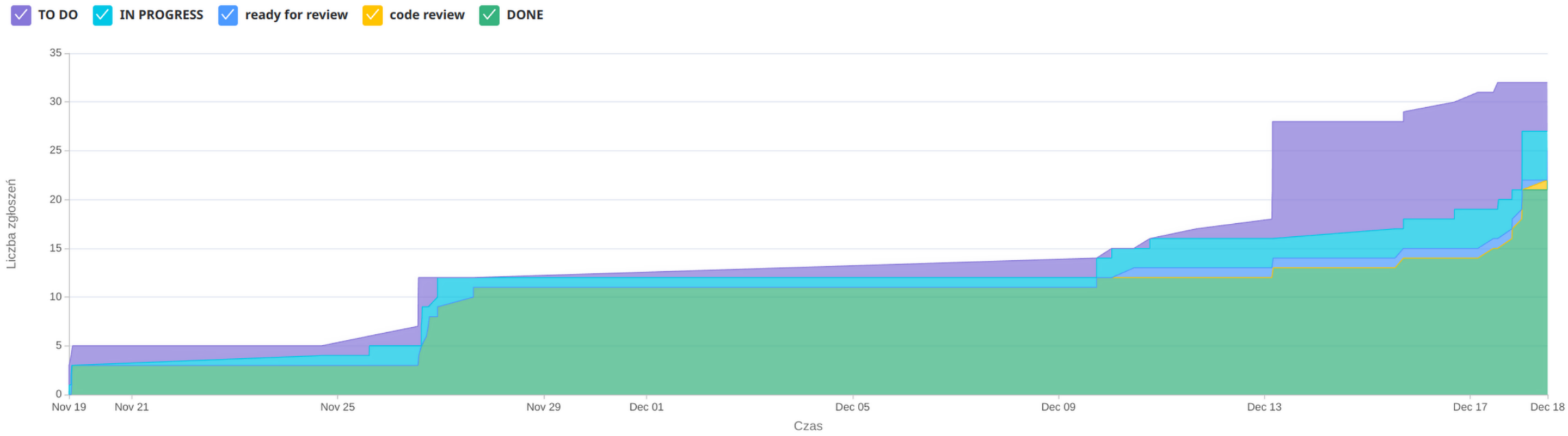
Raport Burnup z Jiry



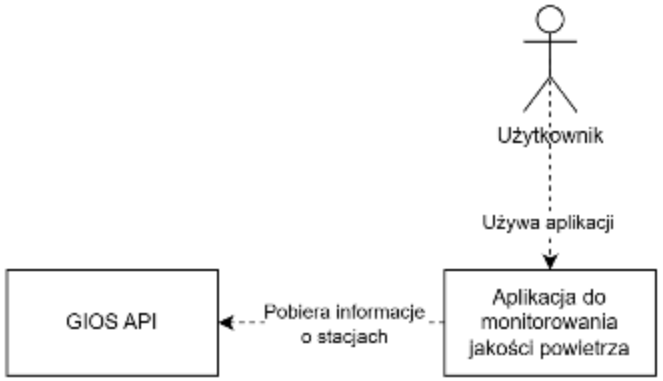
Wykres spalania sprintu z Jiry

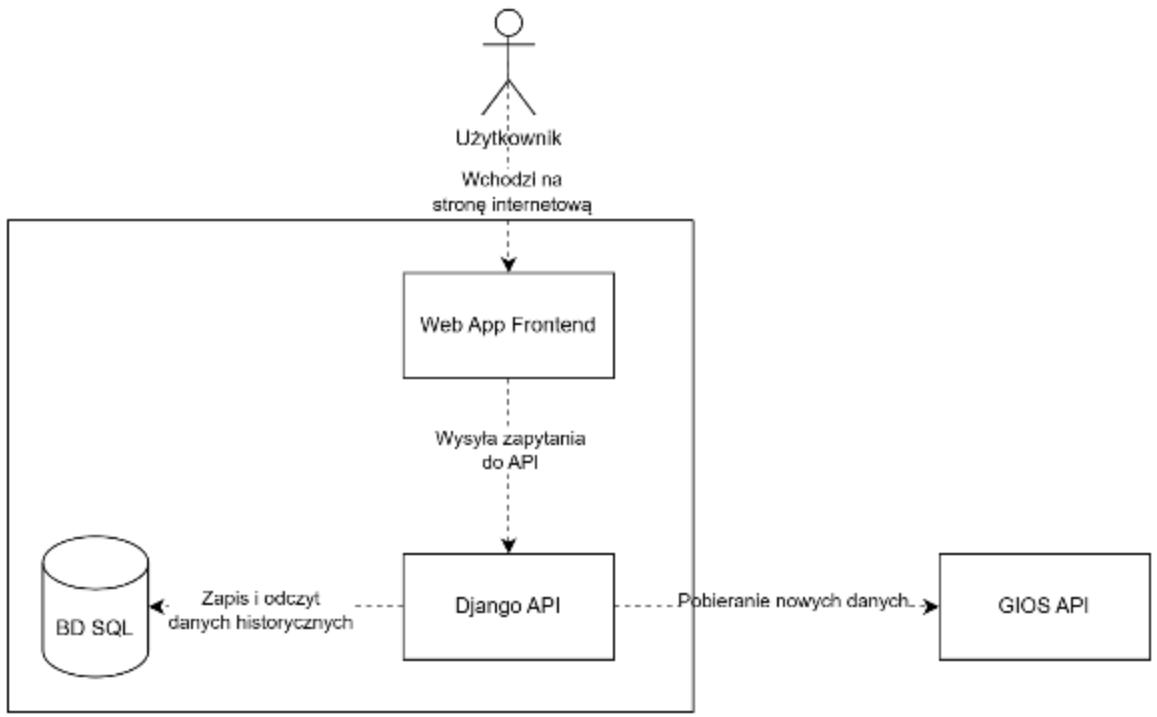


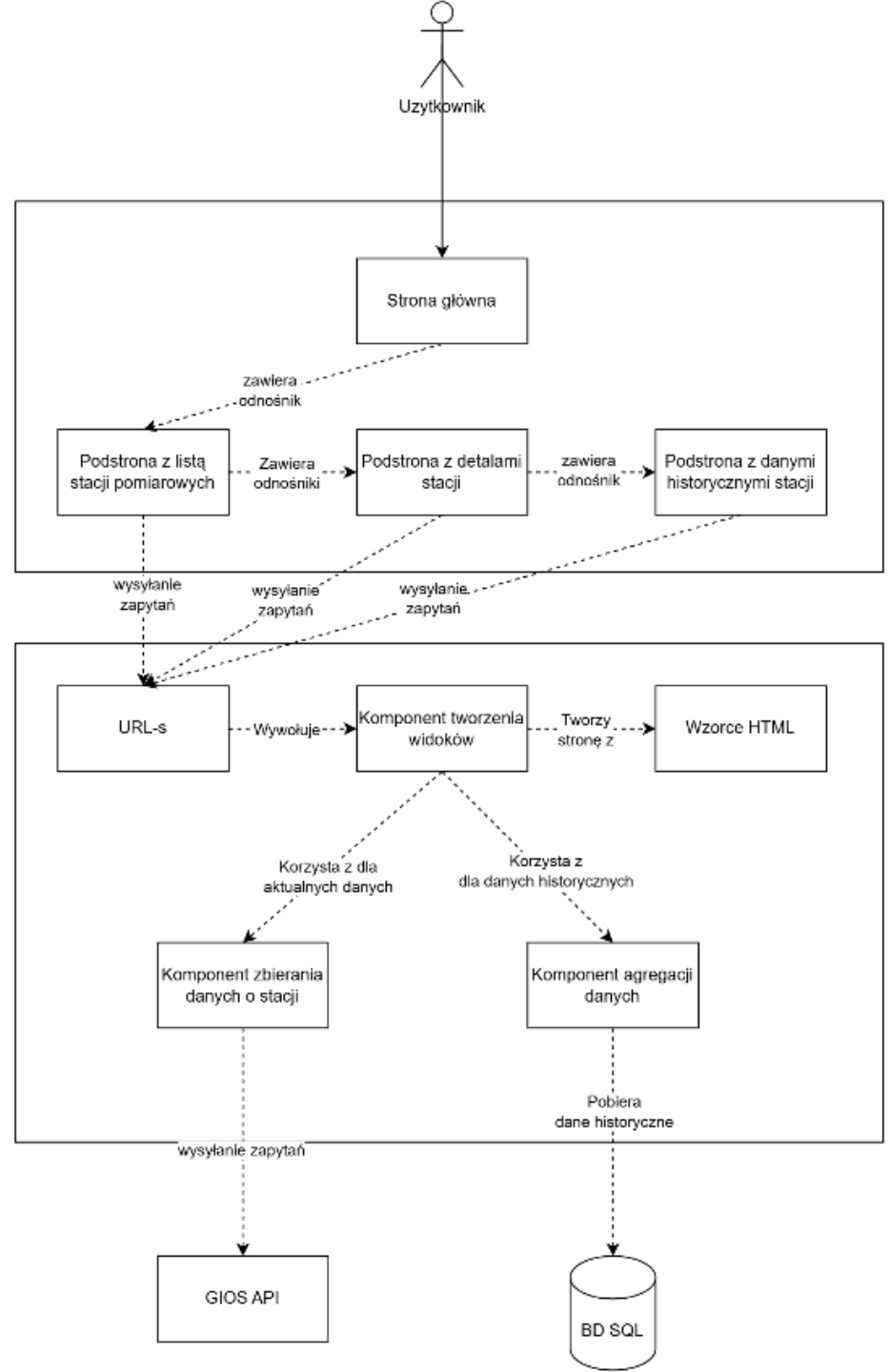
Wykres przepływu skumulowanego z Jiry



Architektura







Model logiczny bazy danych

Technologie

Backend

Frontend

CI/CD

pytest

ATLASSIAN

Jira

Jenkins

Azure

Pipeline z Jenkinsa

Instrukcja zbudowania projektu

Użytkowanie

Uruchamianie aplikacji

- Przejdź do głównego katalogu repozytorium (tam gdzie znajduje się Makefile).
- Uruchom aplikacje poleceniem `make up`.
- Zatrzymaj aplikację (bazę danych) poleceniem `make down`.

Testowanie

- Przejdź do głównego katalogu repozytorium (tam gdzie znajduje się Makefile).
- Zainstaluj zależności poleceniem `make requirements`.
- Uruchom testy poleceniem `make tests`.
- Sprawdz coverage testów poleceniem `make coverage`.
- Usuń pozostałości po testowaniu pokrycia `make clean`.

Lista pozostałych poleceń

- `make migrations` - stosuje migracje (po dodaniu/ zmienieniu modelu)
- `make start_db` - uruchamia kontener z bazą danych
- `make stop_db` - zatrzymuje kontener z bazą danych
- `make run` - uruchamia serwer Django

Testy

```
========================= test session starts ==========================
platform linux -- Python 3.10.12, pytest-8.3.3, pluggy-1.5.0
django: version: 5.1.3, settings: Air_Quality.settings (from ini)
rootdir: /home/slayyymanska/PIS/PIS_Air_Quality/Air_Quality
configfile: pytest.ini
plugins: cov-6.0.0, django-4.9.0, dash-2.18.2
collected 12 items

Air_Quality/gios_api/tests.py ............                      [100%]

---------- coverage: platform linux, python 3.10.12-final-0 ----------
Coverage HTML written to dir htmlcov

Required test coverage of 80% reached. Total coverage: 100.00%

========================== 12 passed in 0.90s ==========================
```

Materiały szkoleniowe

- https://www.youtube.com/watch?v=6YZvp2GwT0A
- https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli
- https://www.digitalocean.com/community/tutorials/how-to-structure-a-terraform-project
- https://docs.ansible.com/ansible/latest/collections/
- https://docs.docker.com/get-started/docker-concepts/building-images/writing-a-dockerfile/
- https://docs.docker.com/reference/cli/docker/image/push/
- https://docs.python.org/3/library/venv.html
- https://www.youtube.com/watch?v=N-RZjp4og28
- https://www.postgresql.org/docs/17/index.html
- https://geshan.com.np/blog/2021/12/docker-postgres/
- https://docs.python.org/3/
- https://docs.djangoproject.com/en/5.1/
- https://www.youtube.com/watch?v=nGIg40xs9e4
- https://docs.streamlit.io/
- https://stackoverflow.com/questions/71201260/django-streamlit-integration
- https://plotly.com/python/

Powyższe linki znajdują się także w pliku docks/educational-materials.md w repozytorium.

GitHub

https://github.com/dominika232323/PIS_Air_Quality

Dokładniej opisane funkcjonalności, instrukcja i stos technologiczny znajdują się w pliku readme.md na githubie.

W folderze docks można obejrzeć wykres architektury oraz logiczny model bazy danych w powiększeniu i lepszej jakości.

Demonstracja