



PIS

Projekt, Etap 2



Wymagania

- Tworzenie fiszek i zestawów
- Nauka
- Przeglądanie fiszek
- Konta
- Statystyki

Tworzenie

Wymagania

1. Użytkownik może tworzyć zestawy fiszek.
2. Użytkownik może tworzyć nowe fiszki, określając pytanie, odpowiedź oraz opcjonalne multimedia.
3. Użytkownik może edytować istniejące fiszki, zmieniając treść pytania, odpowiedzi i multimedia.
4. Użytkownik może dodawać i usuwać poszczególne tagi zestawom fiszek.
5. Użytkownik może usuwać istniejące fiszki.
6. Użytkownik może zmieniać przypisane zestawy dla poszczególnych fiszek.
7. Użytkownik może kopiować i wklejać zarówno poszczególne fiszki, jak i zestawy fiszek.
8. Użytkownik może importować zestawy fiszek z pliku JSON.
9. Użytkownik może eksportować zestawy fiszek do pliku JSON.
10. Użytkownik może oznaczyć zarówno pojedyncze fiszki, jak i zestawy flagą "Ulubione".

Nauka

Wymagania

11. Użytkownik może manualnie przejść przez wszystkie fiszki w zestawie.
12. Użytkownik może uczyć się w turach, przechodząc przez fiszki, oznaczając, czy daną fiszkę się zna, czy nie. W pierwszej turze wyświetlają się wszystkie fiszki w zestawie, a w każdej kolejnej tylko te, które zostały oznaczone jako nieznane.
13. Użytkownik może uczyć się zestawów w trybie spaced repetition, gdzie na podstawie deklaracji znajomości karty, dodaje się ona do kolejki nauki po określonym czasie.

Przeglądanie

Wymagania

- 14. Użytkownik może wyszukać fiszki po pytaniach, odpowiedziach i tagach.
- 15. Użytkownik może filtrować fiszki po tagu, zestawie oraz dacie ostatniego powtórzenia.
- 16. Użytkownik może sortować fiszki po pytaniach, odpowiedziach, tagach, dacie ostatniego powtórzenia oraz po ilości powtórek.

Konta

Wymagania

17. Użytkownik może założyć konto w aplikacji fiszkowej.
18. Użytkownik może zalogować się do aplikacji.

Statystyki

Wymagania

19. Użytkownik może wyświetlić ilość przejranych fiszek.

20. Użytkownik może zobaczyć statystyki dotyczące ilości przejranych fiszek oraz czasu nauki.

21. Użytkownik może zobaczyć statystyki dotyczące poprawności odpowiedzi (np. procent poprawnych odpowiedzi dla każdego zestawu).

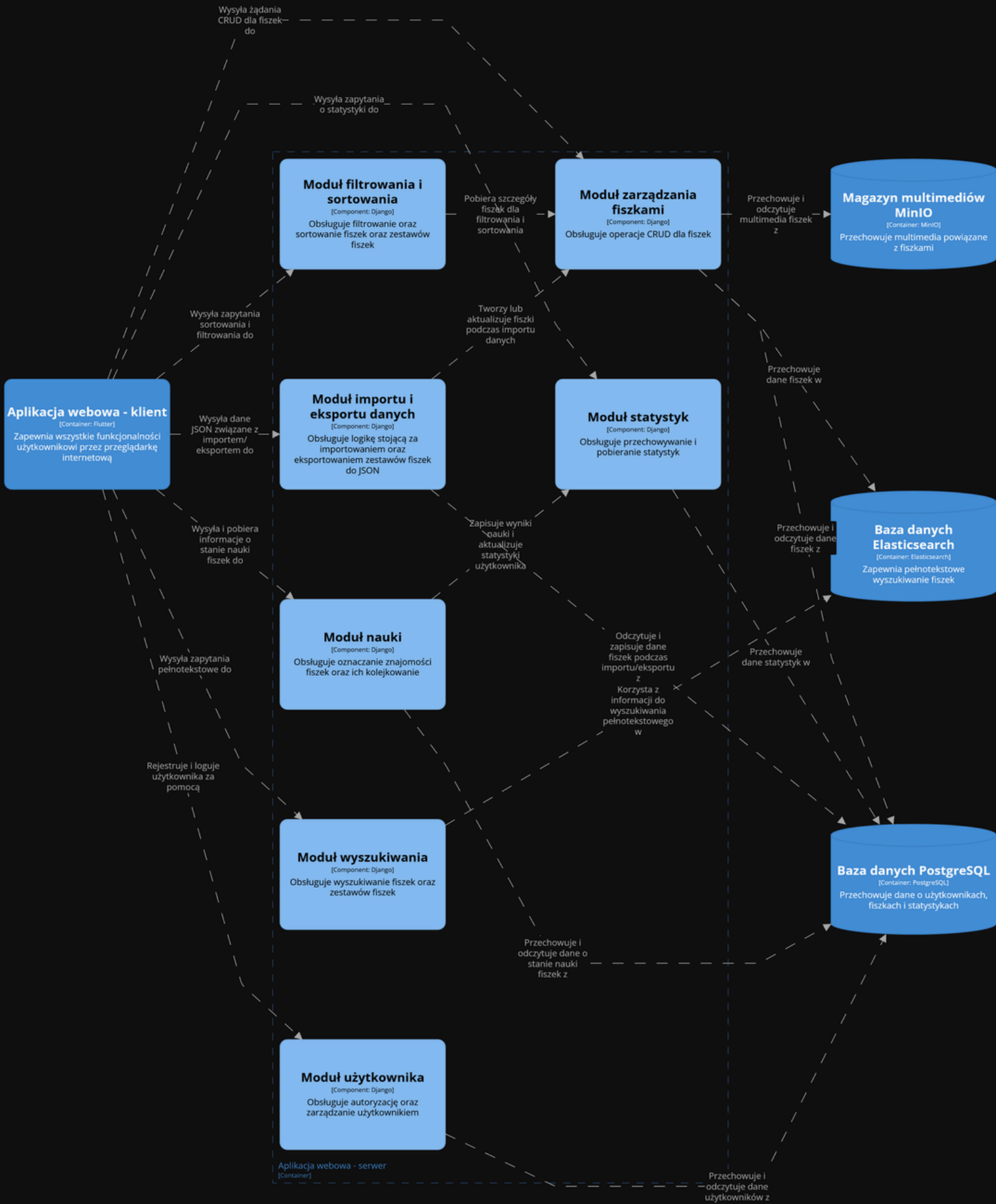


Architektura, poziom 2





Architektura, poziom 3 (backend)



Technologie / narzędzia




- Backend: Django
- Frontend: Flutter
- Bazy danych: PostgreSQL, ElasticSearch, MinIO
- Konteneryzacja: Docker
- CI/CD: Jenkins
- Repozytorium: Github
- Issues: Jira
- Archiwum artefaktów: Nexus

Podział prac

- Angelika Ostrowska - frontend
- Jakub Śledź - wsparcie przy frontendzie
- Michał Brzoza - backend, Jenkins
- Kacper Straszak - bazy danych, konteneryzacja
- Jakub Bąba - sprawy ogólnotechniczne, integracja i spójność rozwiązania, bieżące wsparcie

Źródła, z których korzystamy



- <https://github.com/ashgole/Django-Flutter-CRUD-operations-DoctorApp>
- <https://youtu.be/qcJZN1pvG6A>
- <https://youtu.be/sm1makevMWk>
- https://youtube.com/playlist?list=PLzMxCBgfZo4-knQWKGK2IC49Q_5AnQrFpzv
- <https://docs.flutter.dev/cookbook/forms/validation>
- <https://youtu.be/U7z5leWuaLI?si=NXA27fXMBR3cVCst>
- <https://min.io>
- <https://elastic.co>
- <https://hub.docker.com>