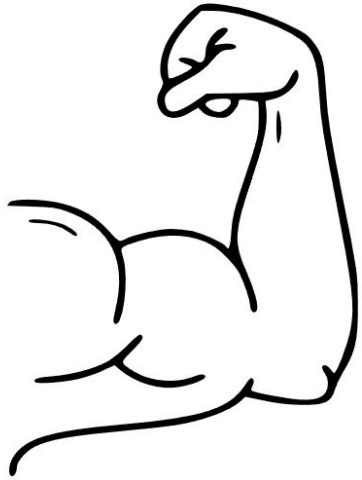


Zarządzanie Aplikacjami Internetowymi



Gymshare

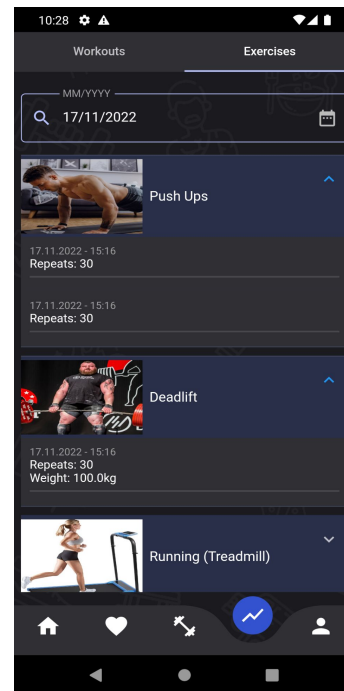
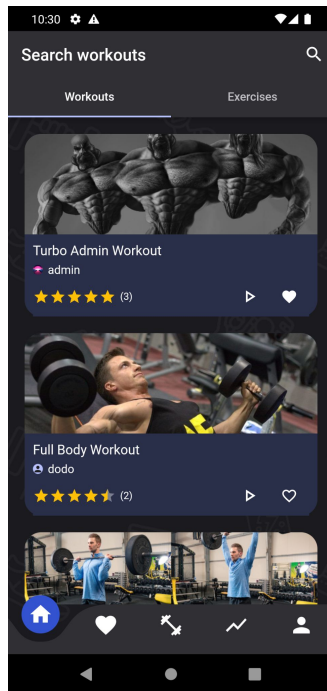
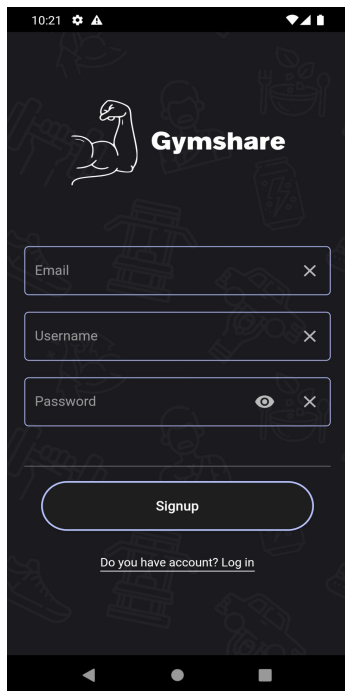
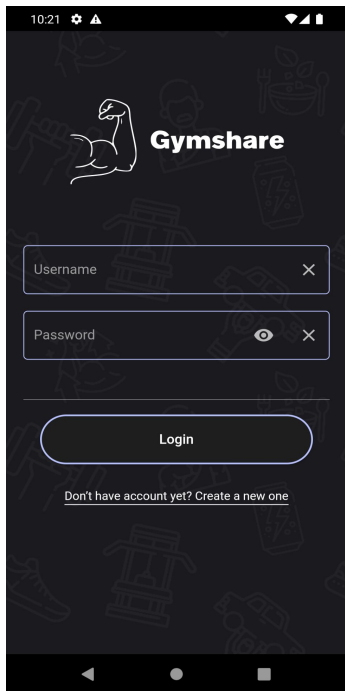
Patryk Jędrzejewski
Filip Kaczmarek
Andrzej Kapczyński

Opis aplikacji

- Ułatwia zarządzanie treningiem,
- Baza gotowych ćwiczeń,
- Możliwość stworzenia własnego planu treningowego,
- Odtwarzanie treningu ułatwia śledzenie postępów,
- Program ma służyć użytkownikom lokalnej siłowni znajdującej się przy ul. Wioślarskiej w Poznaniu.

Aplikacja została zaprojektowana jako aplikacja mobilna, lecz na potrzeby przedmiotu została przebudowana do aplikacji internetowej. Stos technologiczny: Python, Django, Dart, Flutter, PostgreSQL

Przykładowe widoki aplikacji



Wymagania funkcjonalne

Niezałogowani oraz nowi użytkownicy aplikacji mają możliwość:

- U1. Stworzenia nowego konta.
- U2. Zresetowania hasła do istniejącego już konta.
- U3. Zalogowania się do aplikacji, podając adres e-mail oraz hasło.

Wymagania funkcjonalne

Zalogowany użytkownik ma możliwość:

- Z1. Przeglądania planów treningowych.
- Z2. Tworzenia planów treningowych.
- Z3. Przeglądania listy ćwiczeń.
- Z4. Przeglądania prywatnych statystyk.
- Z5. Edycji danych w profilu.
- Z6. Zmiany hasła do konta.
- Z7. Edycji swoich planów treningowych.
- Z8. Usuwania swoich planów treningowych.
- Z9. Dodawania planów treningowych do listy ulubionych.
- Z10. Oceniania planów treningowych.
- Z11. Uruchamiania planów treningowych..
- Z12. Wylogowania z konta.

Wymagania pozafunkcjonalne

- Aplikacja powinna uruchamiać się na przeglądarkach internetowych Chrome, Opera, Safari, Firefox, Edge.
- Aplikacja powinna mieć dostęp do Internetu.
- Hasła użytkowników w bazie danych powinny być szyfrowane.
- Przewidywana maksymalna liczba zalogowanych użytkowników na raz: 50.
- Maksymalna ilość zarejestrowanych użytkowników: 500.

Specyfikacja serwera

Starter

Trial

Fiddle with the platform sans credit card.

512 MB RAM

and shared CPU / container

1 GB Disk

Split between your services

\$5 or 500 hours of usage

Deploys shut off when first limit is reached

Community Support

Documentation & Discord Community

Get Started

Developer

Usage

Ditch the limits. Still free under \$5/mo.

Up to 8 GB RAM

and shared CPU / container

Up to 100 GB Disk

Split between your services

Unlimited execution time

Deploys remain online indefinitely

Prepaid or Usage-based

Multiple payment options to fit your needs

Choose Plan

Team

\$20 / seat

Scale up and outship your competitors.

Up to 32 GB RAM

and up to 32 vCPU / container

Up to 2 TB Disk

Split between your services

Shared Projects

Members can access all team projects

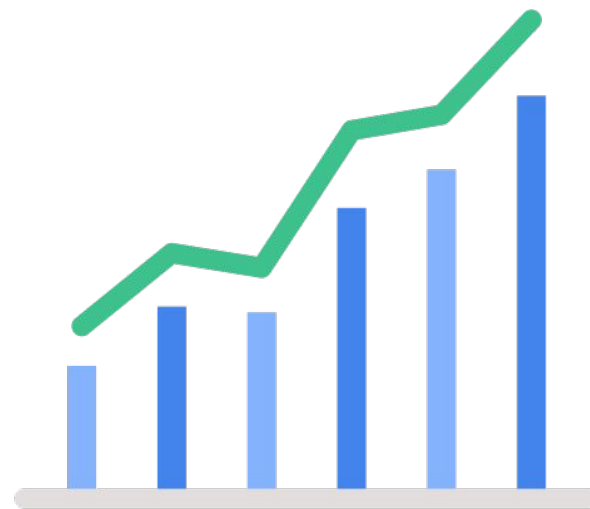
Direct Support Line

Dedicated support channel with your team

Create a Team

Testy

1. Testy aplikacji serwerowej za pomocą narzędzia JMeter.
2. Testy integracyjne aplikacji internetowej.



Funkcje do testowania i oczekiwane czasy odpowiedzi aplikacji serwerowej.

Nazwa funkcji	Czas odpowiedzi [s]
Tworzenie konta	2
Logowanie na konto użytkownika	2
Utworzenie treningu	4
Wyświetlenie listy treningów	2
Przetwarzanie zakończenia treningu	1
Wystawienie oceny treningu	1
Czas generowania statystyk spalonych kalorii	3
Czas generowania statystyk odbytych ćwiczeń	3



Python



Scenariusze testowania aplikacji internetowej

Tworzenie konta

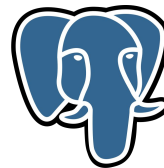
1. Użytkownik przechodzi do strony rejestracji.
 2. Użytkownik wypełnia formularz a następnie go zatwierdza.
 3. Użytkownik loguje się do aplikacji po uzyskaniu informacji o założeniu konta.
-
- Wyświetlenie treningów
 - Wyświetlenie statystyk spalonych kalorii
 - Wyświetlenie statystyk wykonanych ćwiczeń



Populowanie bazy danych

Baza danych została wypełniona testowymi danymi, tak aby spełniać przybliżone warunki przy 500 zarejestrowanych użytkownikach. Dodano:

- 500 użytkowników,
- 1500 treningów,
- 250 ćwiczeń,
- 30 tysięcy statystyk odnośnie spalonych kalorii,
- 170 tysięcy statystyk odnośnie wykonanych ćwiczeń.



PostgreSQL

Wybrana baza danych spełniła oczekiwania co do pomieszczenia danych wygenerowanych przez 500 użytkowników.

Testowanie aplikacji serwerowej

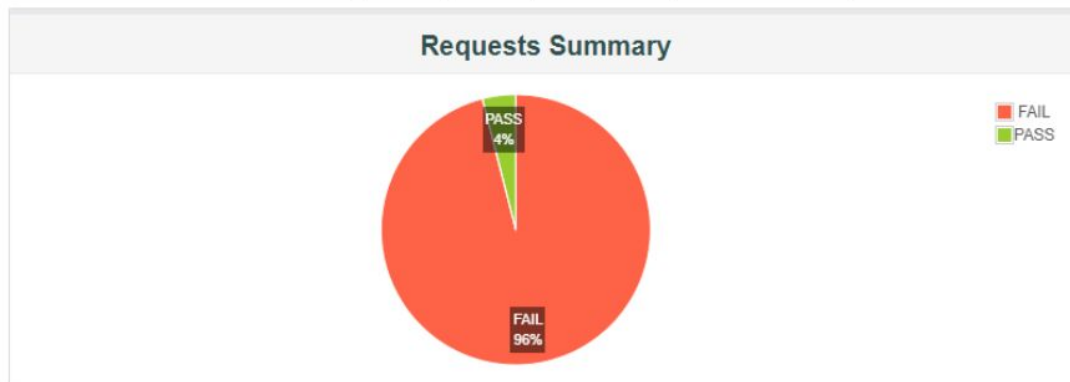
- JMeter
- Liczba aktywnych użytkowników = 50

Testy podzielono na:

- Warunki normalne
- Stress test, Crash test

Tworzenie konta

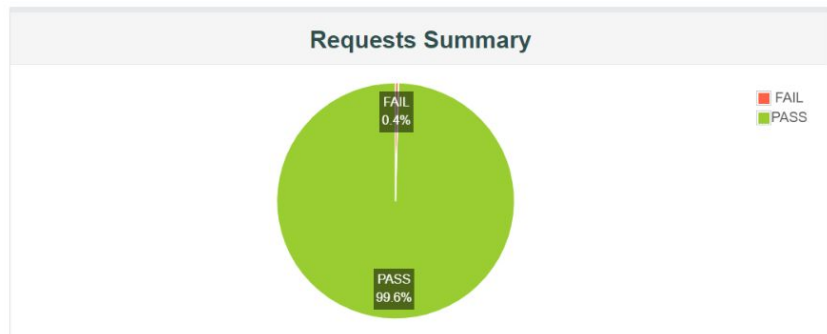
i. Warunki normalne (maksymalny czas odpowiedzi: **2s**)



Wykres przedstawia zapytania, które uzyskały odpowiedzi poniżej zakładanego czasu.

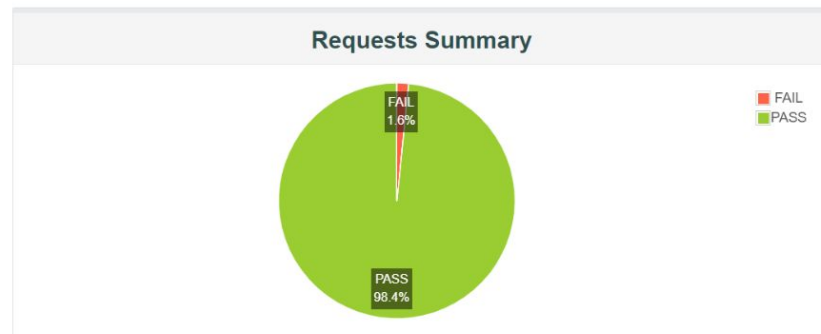
Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	50	48	7635.54	1319	13717

Tworzenie konta



Wykres przedstawia zapytania, które uzyskały odpowiedzi.

Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	500	2	63149.16	1012	300510

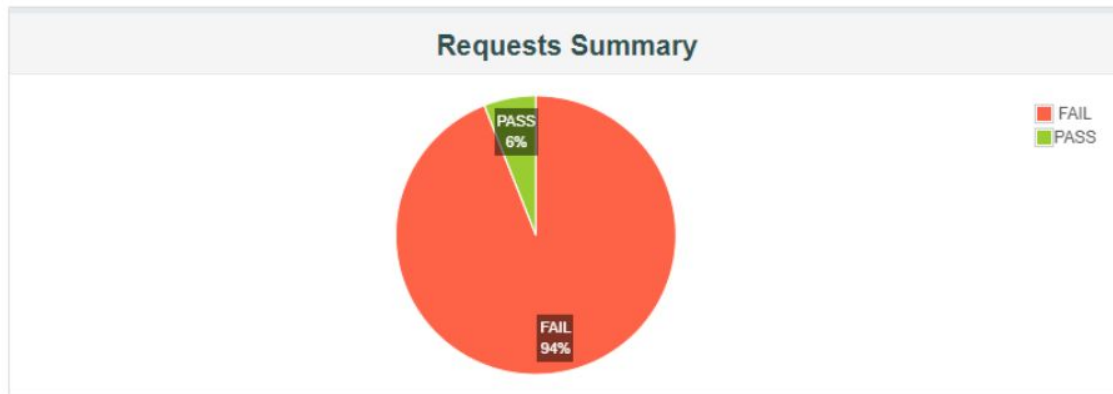


Wykres przedstawia zapytania, które uzyskały odpowiedzi.

Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	1000	16	125273	1435	305713

Logowanie na konto użytkownika

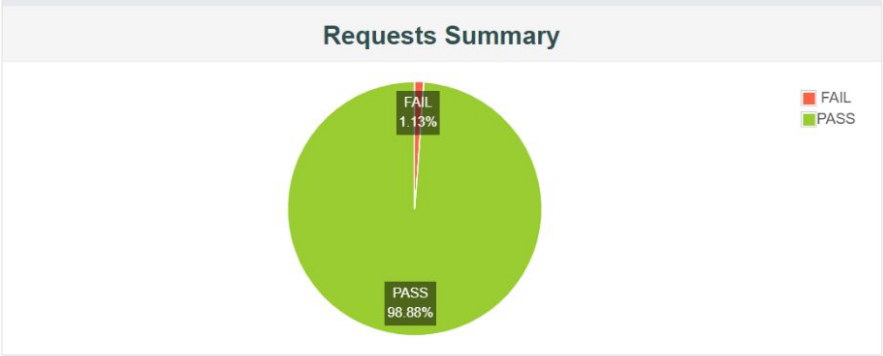
i. Warunki normalne (maksymalny czas odpowiedzi: **2s**)



Wykres przedstawia zapytania, które uzyskały odpowiedzi poniżej zakładanego czasu.

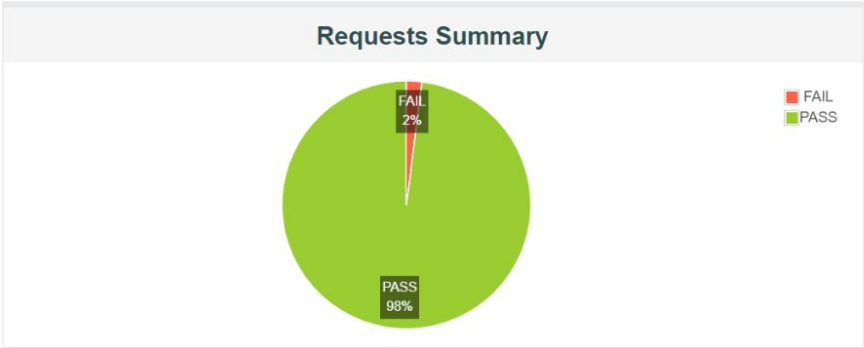
Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	50	47	7367.22	1519	13264

Logowanie na konto użytkownika



Wykres przedstawia zapytania, które uzyskały odpowiedzi.

Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	800	9	95977	1635	300577



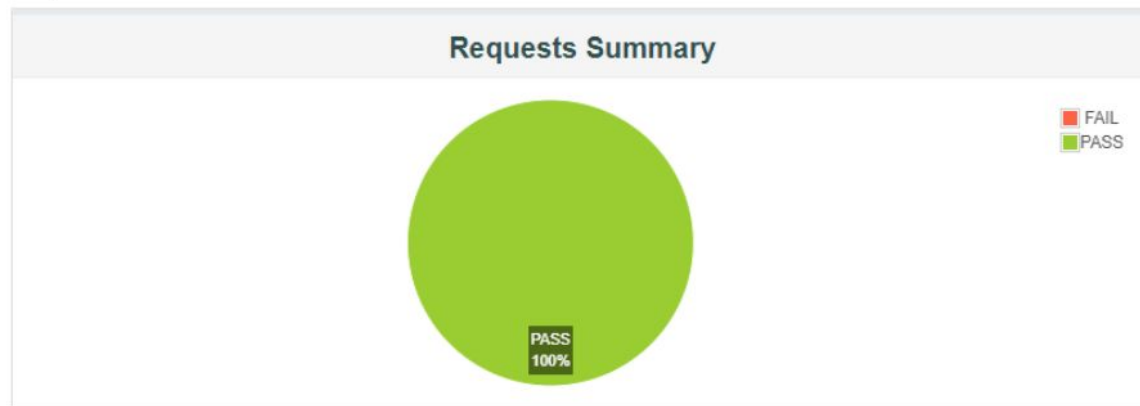
Wykres przedstawia zapytania, które uzyskały odpowiedzi.

Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	1000	20	120804	1107	300534

Utworzenie treningu

Maksymalny czas odpowiedzi: **4s**

Raport JMeter



Wykres przedstawia zapytania, które uzyskały odpowiedzi poniżej zakładanego czasu.

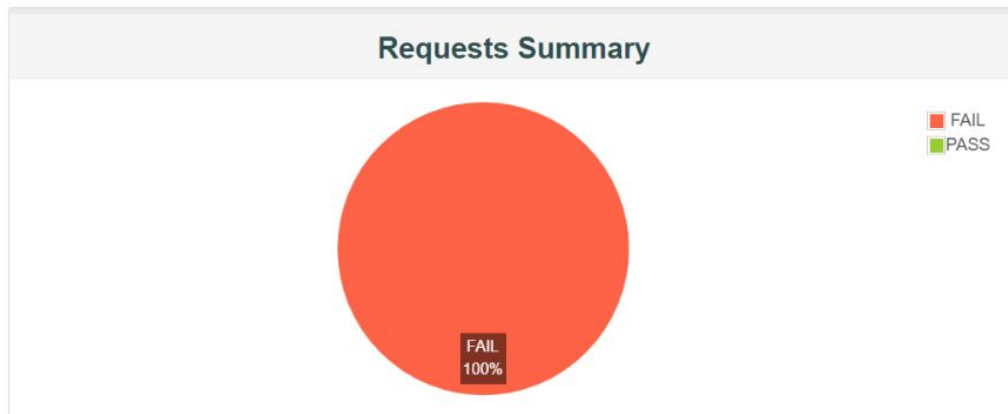
Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	50	0	1364.36	1143	1601

Utworzenie treningu

Funkcja spełnia postawione wymagania czasowe, co więcej są one lepsze od zakładanych prawie 4 krotnie. Próby testowania tej funkcji na 500 i 1000 aktywnych użytkownikach przyniosły także dobre rezultaty przez co zaprzestano dalszych testów z powodu obaw o naliczenie opłat jakie występują przy wybranym planie hostingu.

Wyświetlenie listy treningów

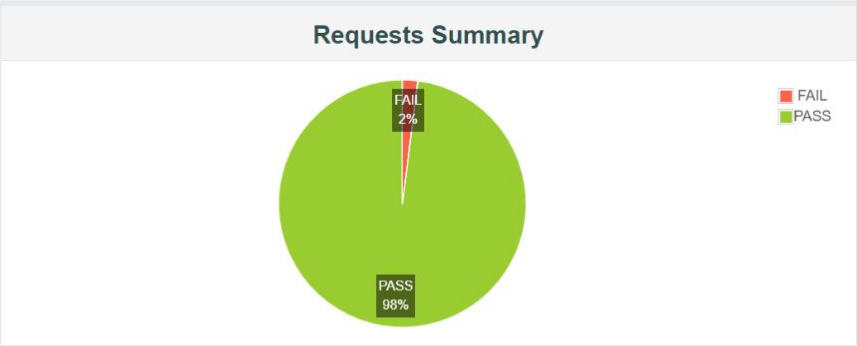
i. Warunki normalne (maksymalny czas odpowiedzi: **2s**)



Wykres przedstawia zapytania, które uzyskały odpowiedzi poniżej zakładanego czasu.

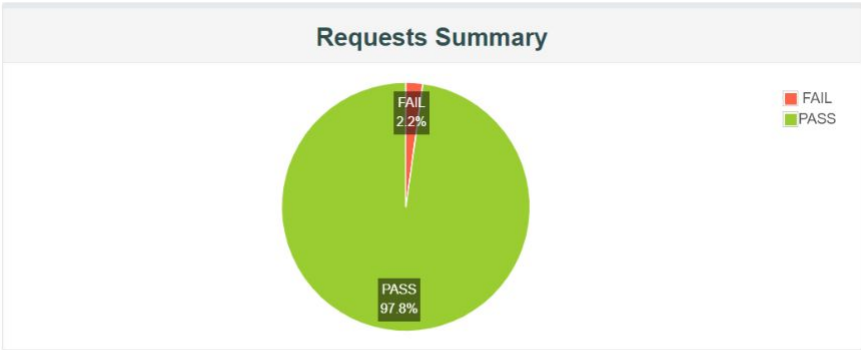
Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	50	50	19372.26	2252	36200

Wyświetlenie listy treningów



Wykres przedstawia zapytania, które uzyskały odpowiedzi.

Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	350	7	107731	2416	302584

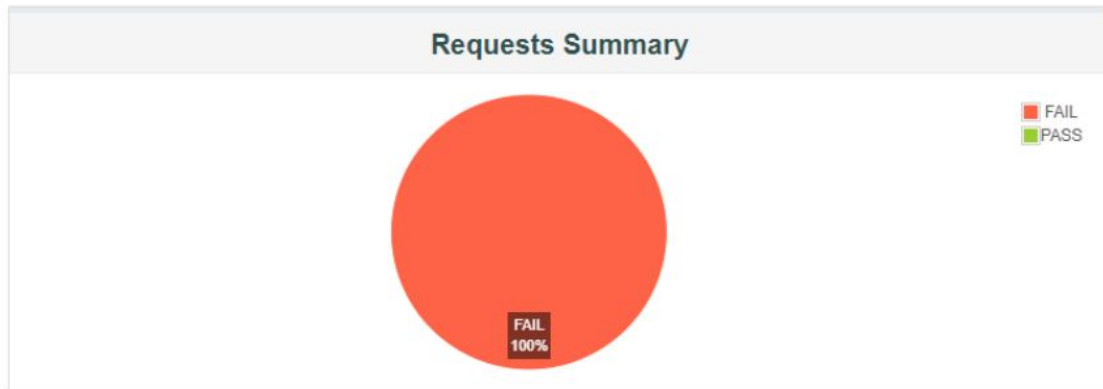


Wykres przedstawia zapytania, które uzyskały odpowiedzi.

Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	500	11	149471	2641	302361

Przetwarzanie zakończenia treningu

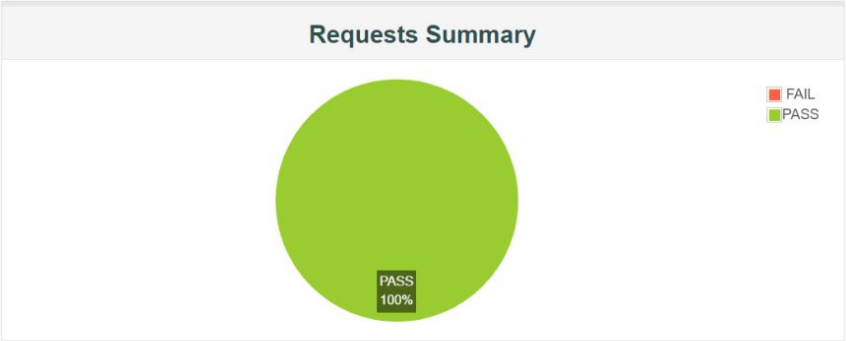
i. Warunki normalne (maksymalny czas odpowiedzi: **1s**)



Wykres przedstawia zapytania, które uzyskały odpowiedzi poniżej zakładanego czasu.

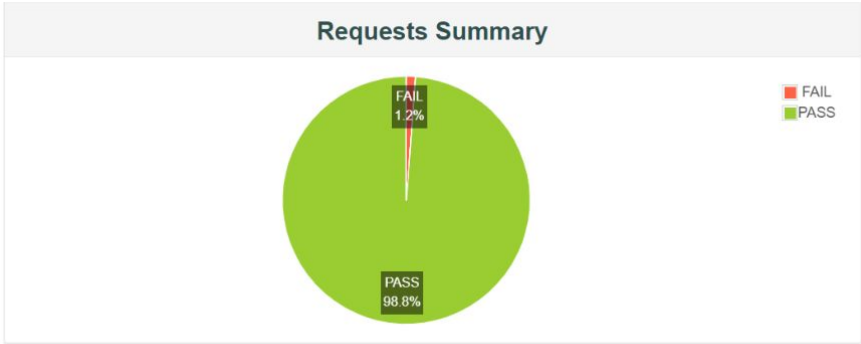
Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	50	50	3101.34	1190	5021

Przetwarzanie zakończenia treningu



Wykres przedstawia zapytania, które uzyskały odpowiedzi.

Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	1500	0	73401.18	14961	131842



Wykres przedstawia zapytania, które uzyskały odpowiedzi.

Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	2000	24	82568	2907	304786

Wystawianie oceny treningu

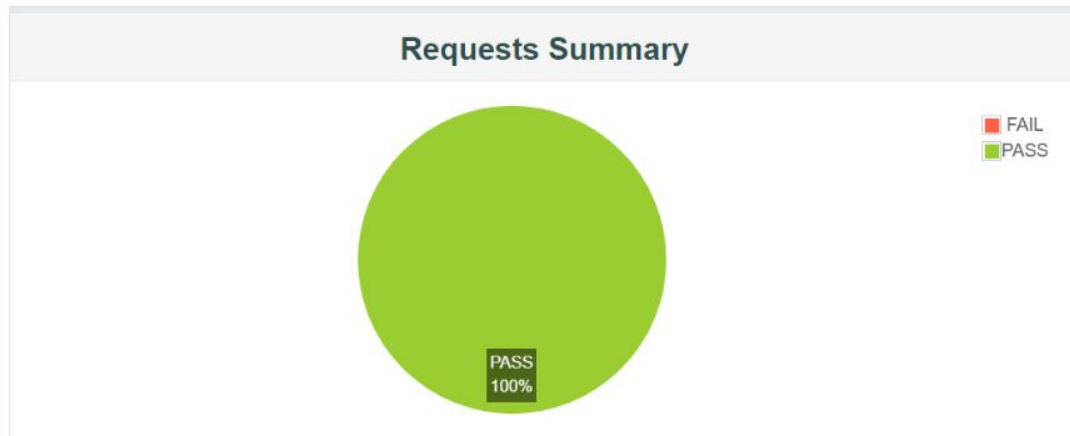
- i. Warunki normalne (maksymalny czas odpowiedzi: **1s**)



Wykres przedstawia zapytania, które uzyskały odpowiedzi poniżej zakładanego czasu.

Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	50	50	2219.38	1975	2495

Wystawianie oceny treningu

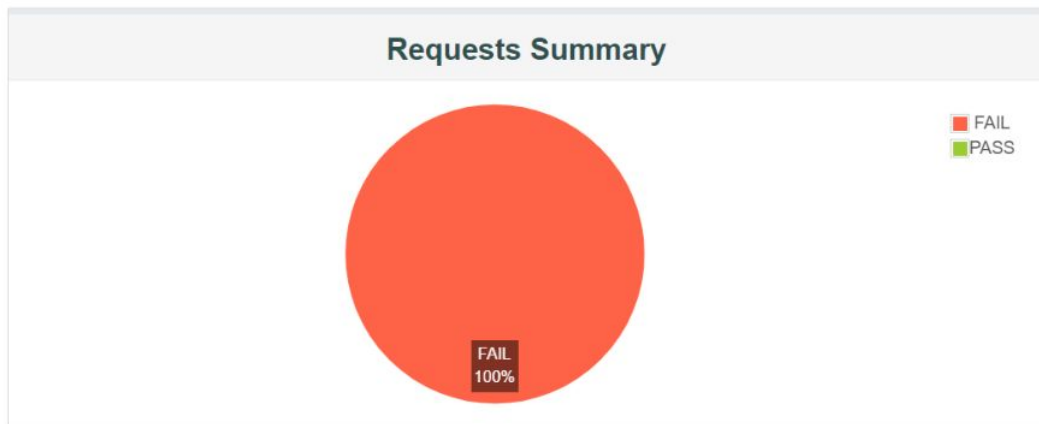


Wykres przedstawia zapytania, które uzyskały odpowiedzi.

Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	2000	0	13231.66	3535	22981

Czas generowania statystyk spalonych kalorii

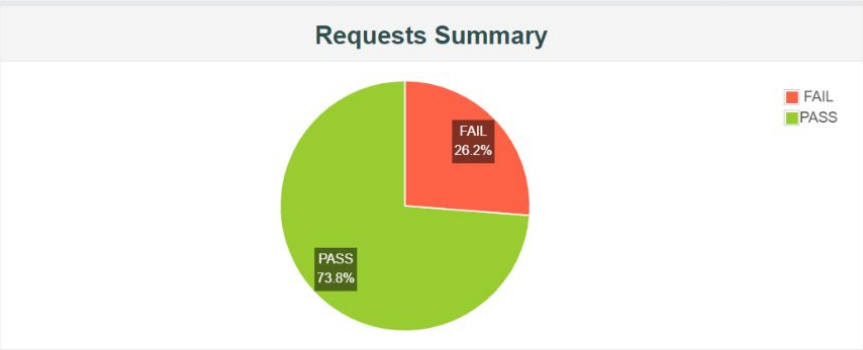
i. Warunki normalne (maksymalny czas odpowiedzi: **3s**)



Wykres przedstawia zapytania, które uzyskały odpowiedzi poniżej zakładanego czasu.

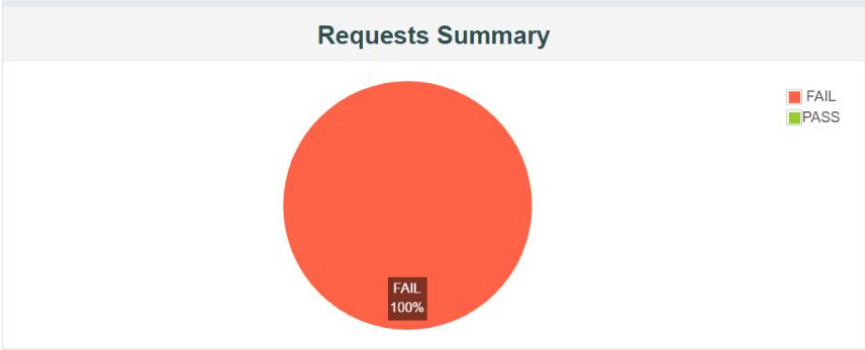
Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	50	50	3900.30	3877	3949

Czas generowania statystyk spalonych kalorii



Wykres przedstawia zapytania, które uzyskały odpowiedzi.

Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	2000	524	23298.40	17315	29082

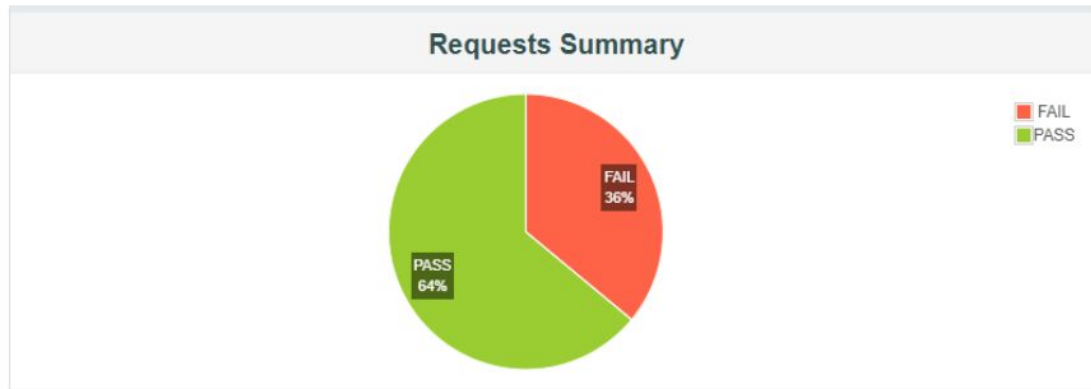


Wykres przedstawia zapytania, które uzyskały odpowiedzi.

Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	3000	3000	21017.92	19042	21961

Czas generowania statystyk odbytych ćwiczeń

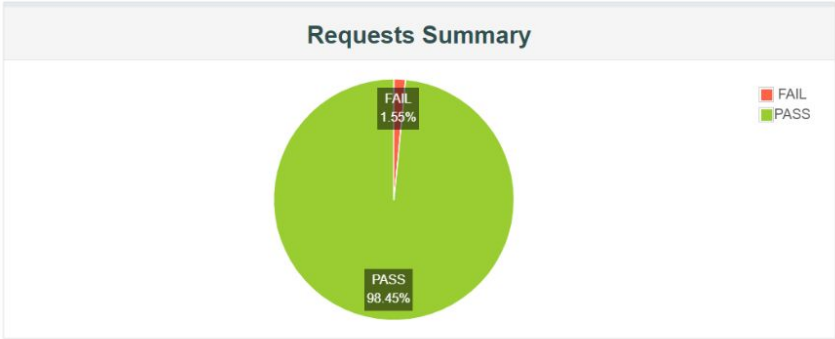
i. Warunki normalne (maksymalny czas odpowiedzi: **3s**)



Wykres przedstawia zapytania, które uzyskały odpowiedzi poniżej zakładanego czasu.

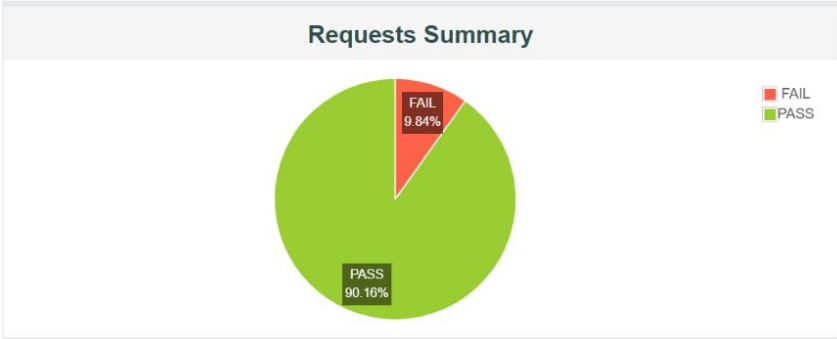
Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	50	18	2586.68	1197	3977

Czas generowania statystyk odbytych ćwiczeń



Wykres przedstawia zapytania, które uzyskały odpowiedzi.

Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	2000	31	76261.59	4192	307149



Wykres przedstawia zapytania, które uzyskały odpowiedzi.

Żądania	Wykonania		Czas odpowiedzi [ms]		
	Próbki	Porażki	Średnia	Min	Max
Całkowicie	2500	246	80787.31	5055	315960

Testy integracyjne UI

Tworzenie konta

Czas [s]		
Średnia	Min	Max
12.20	11.82	13.18

Wyświetlenie treningów

Czas [s]		
Średnia	Min	Max
23.15	22.89	23.65

Wyświetlenie statystyk spalonych kalorii

Czas [s]		
Średnia	Min	Max
46.12	45.81	46.78

Wyświetlenie statystyk wykonanych ćwiczeń

Czas [s]		
Średnia	Min	Max
41.25	41.04	42.01

Audyt aplikacji - Google Lighthouse

desktop

mobile



METRICS

● First Contentful Paint
0.2 s

▲ Total Blocking Time
470 ms

▲ Speed Index
2.5 s

METRICS

▲ First Contentful Paint
4.2 s

▲ Total Blocking Time
1,260 ms

▲ Speed Index
10.2 s

Optymalizacja

Po wykonanych testach zauważono wąskie gardła aplikacji oraz pomysły na ich usunięcie/poprawienie:

- Wąskie gardło: Odpowiedź otrzymywana po wykonaniu zapytania na endpoint odpowiedzialny za wyświetlenie planów treningowych nie jest stronicowana co w przypadku dużej ilości danych w bazie danych wpływa negatywnie na wydajność aplikacji serwerowej.

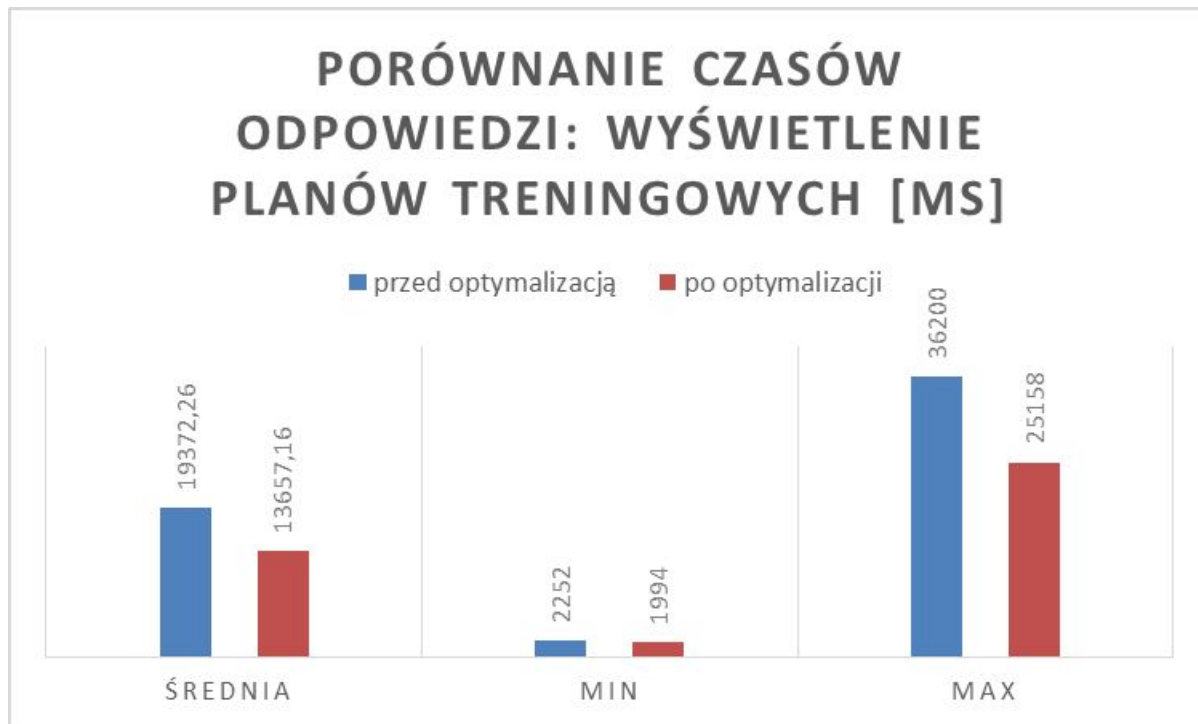
Rozwiązanie: Dodanie paginacji.

- Wąskie gardło: Komenda uruchamiająca aplikację serwerową nie jest sparametryzowana parametrem “workers”, który określa liczbę procesów.

Rozwiązanie: Dobranie optymalnej wartości tego parametru.

- Pomysł: Zaktualizowanie Fluttera do najnowszej wersji 3.10, która zwiększa wydajność aplikacji w przeglądarce internetowej.
- Pomysł: Aktualizacja bibliotek aplikacji internetowej do najnowszych wersji.

Dodanie paginacji



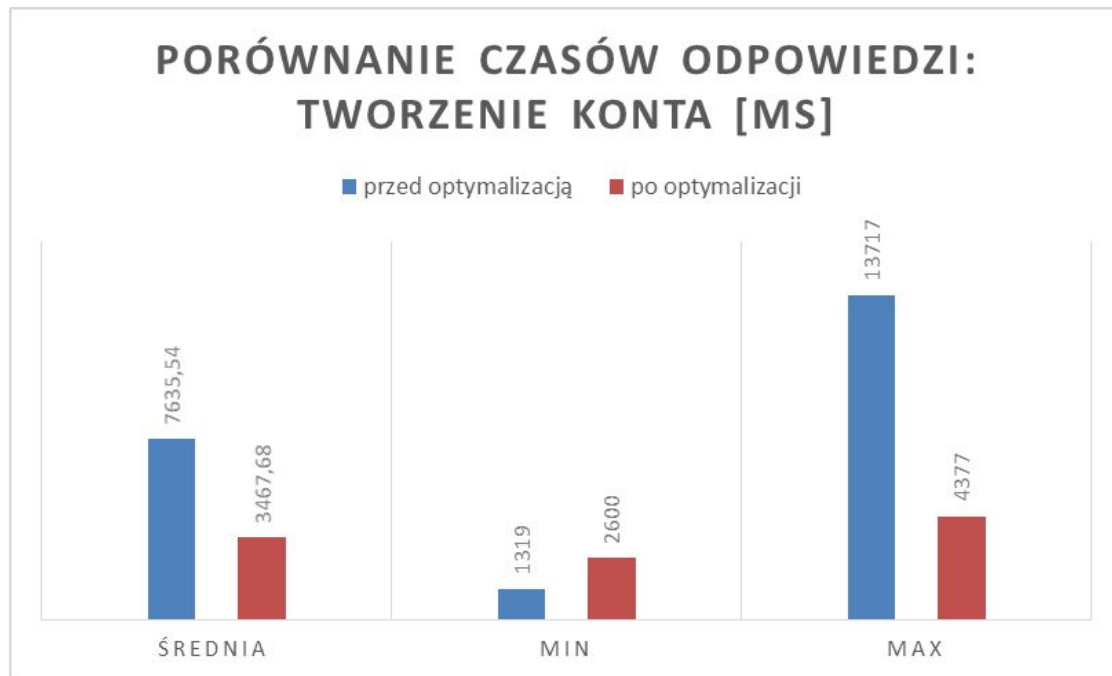
Parametryzacja komendy uruchamiającej aplikację serwerową

Według dokumentacji Gunicorn, parametr workers powinien zostać obliczony ze wzoru:

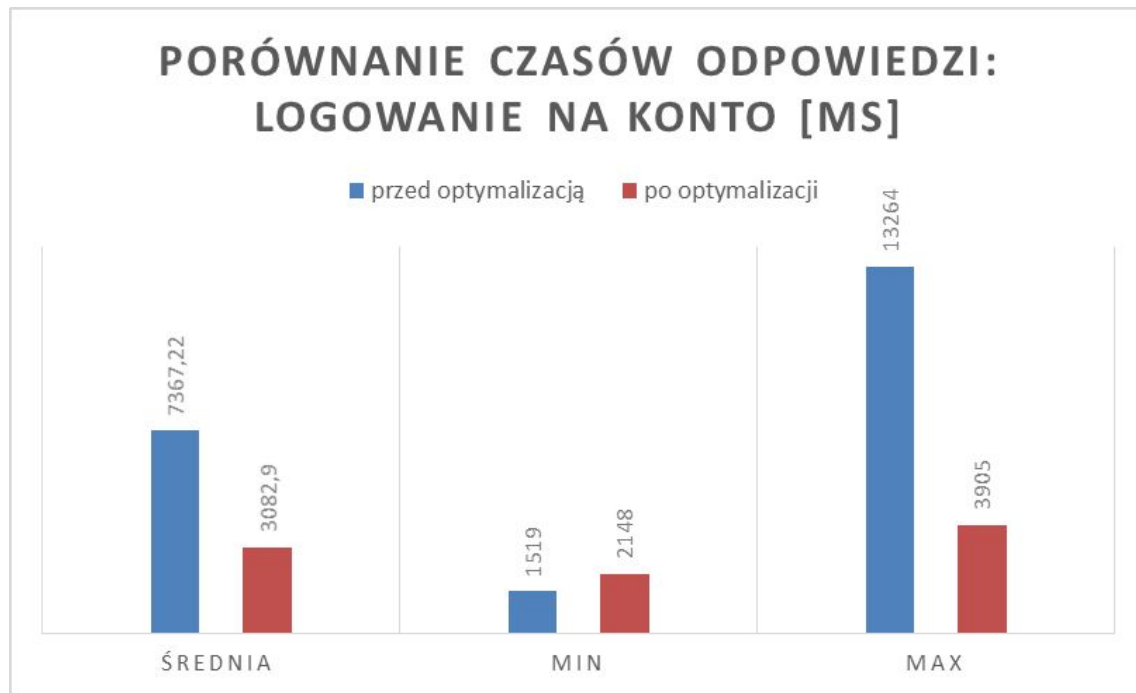
$$(2 * \text{liczba_rdzeni}) + 1$$

Podstawiając dane do wzoru otrzymujemy: $(2 * 8) + 1 = 17$.

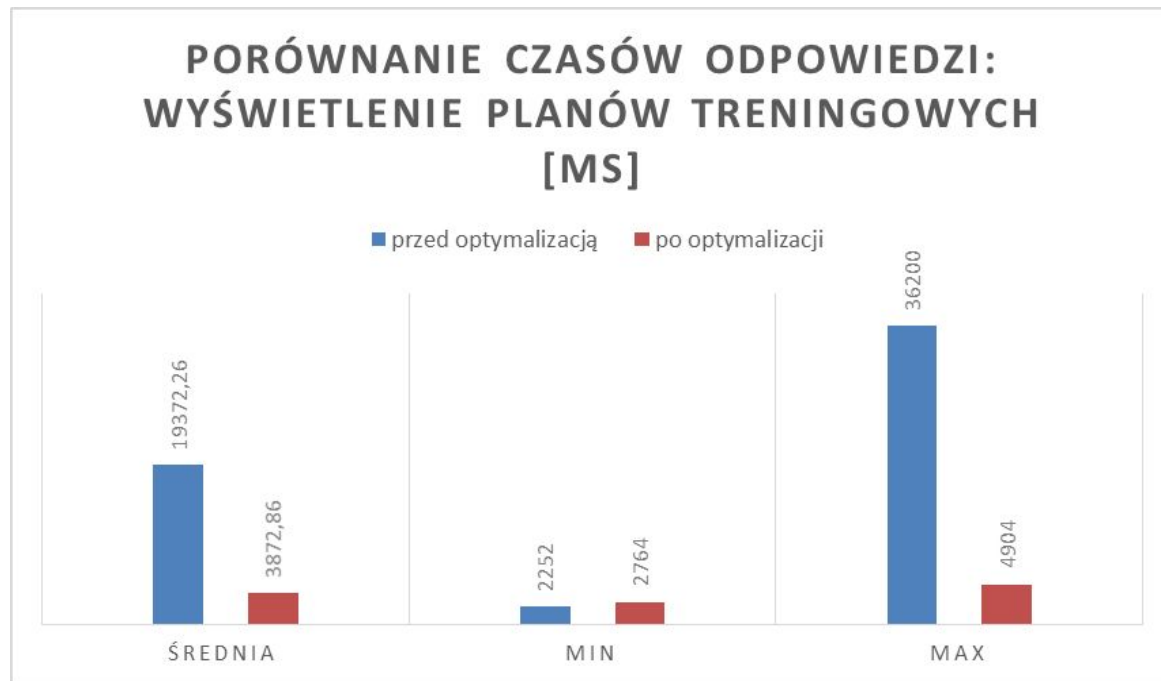
Wyniki optymalizacji aplikacji serwerowej (dla 50 aktywnych użytkowników):



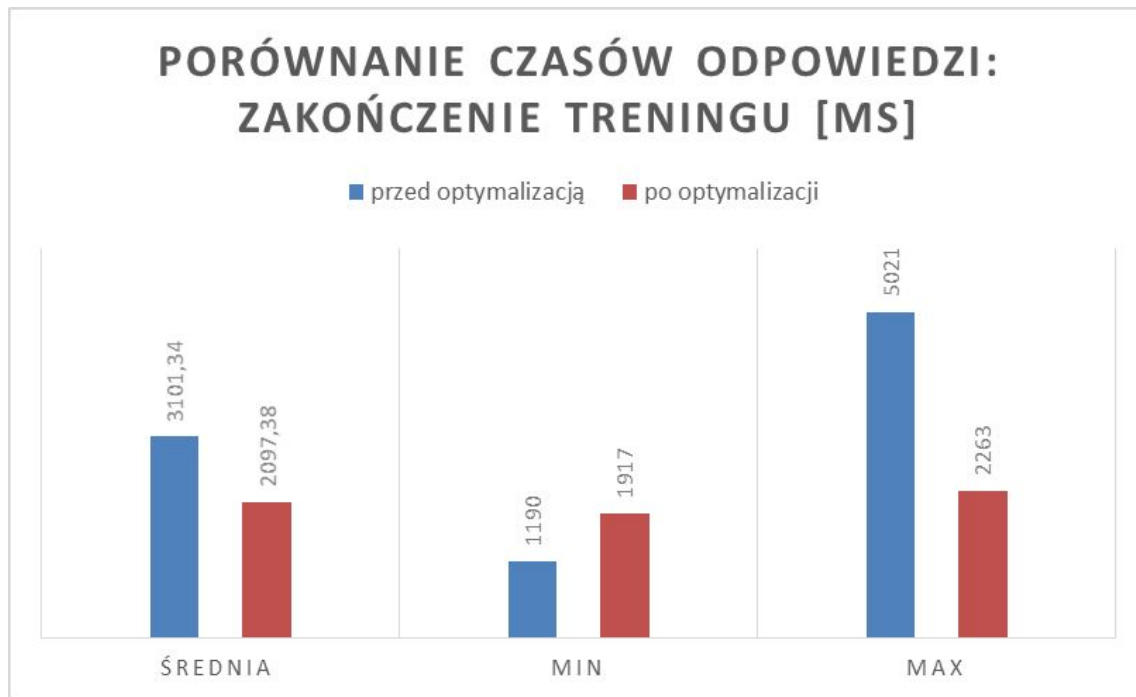
Wyniki optymalizacji aplikacji serwerowej (dla 50 aktywnych użytkowników):



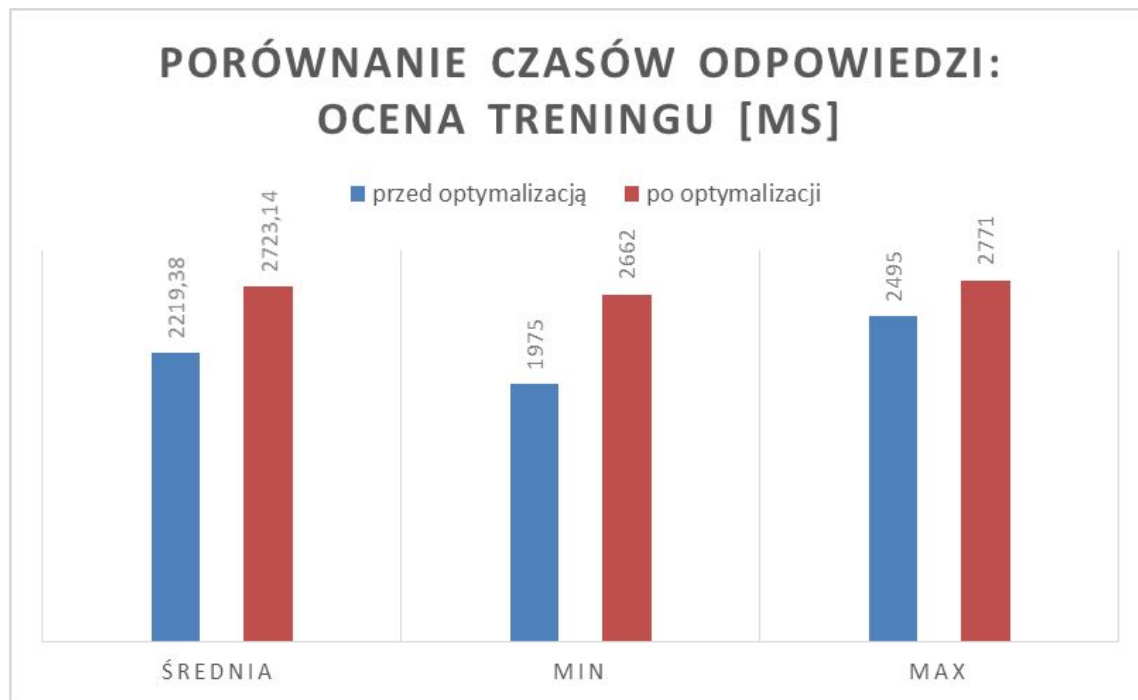
Wyniki optymalizacji aplikacji serwerowej (dla 50 aktywnych użytkowników):



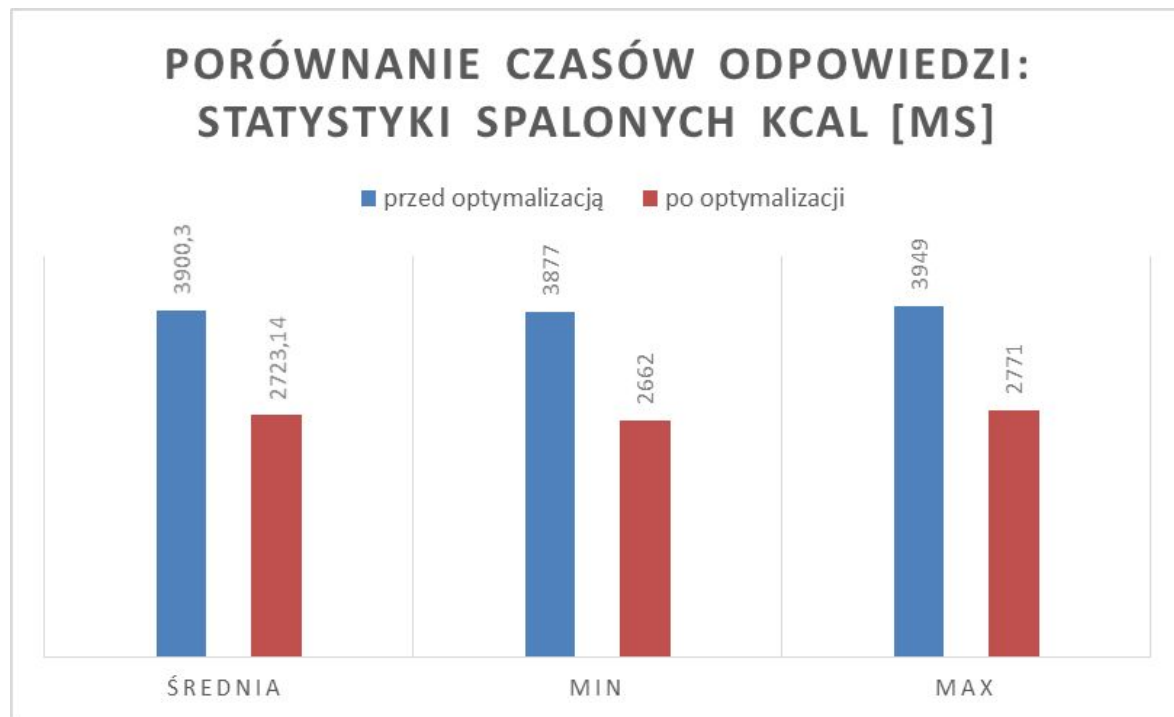
Wyniki optymalizacji aplikacji serwerowej (dla 50 aktywnych użytkowników):



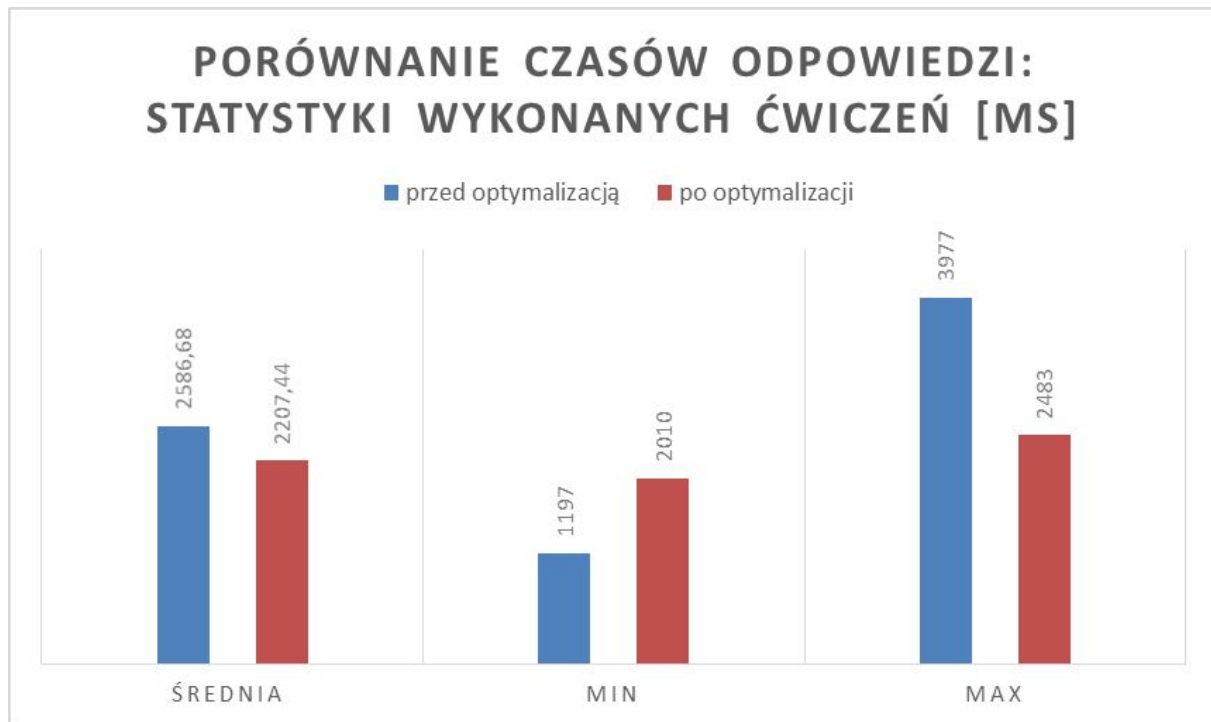
Wyniki optymalizacji aplikacji serwerowej (dla 50 aktywnych użytkowników):



Wyniki optymalizacji aplikacji serwerowej (dla 50 aktywnych użytkowników):



Wyniki optymalizacji aplikacji serwerowej (dla 50 aktywnych użytkowników):



Wyniki testów integracyjnych UI po optymalizacji

Tworzenie konta

Czas [s]		
Średnia	Min	Max
12.05	11.76	12.35

Wyświetlenie treningów

Czas [s]		
Średnia	Min	Max
17.99	17.16	19.43

Wyświetlenie statystyk spalonych kalorii

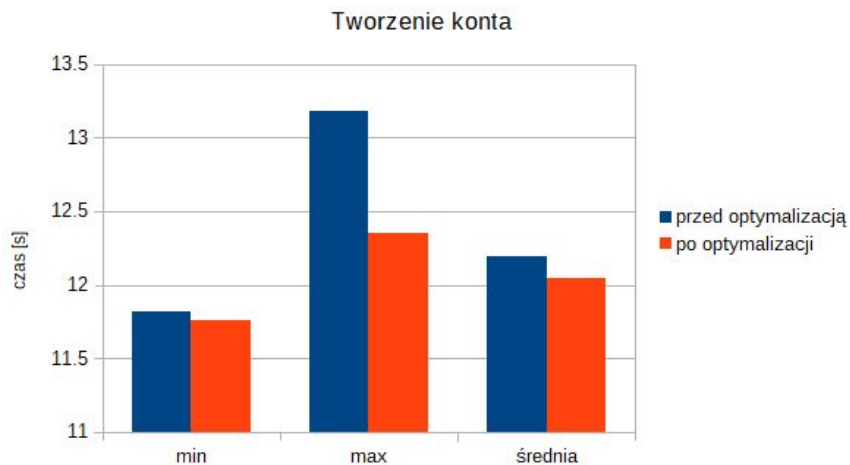
Czas [s]		
Średnia	Min	Max
12.56	12.24	12.91

Wyświetlenie statystyk wykonanych ćwiczeń

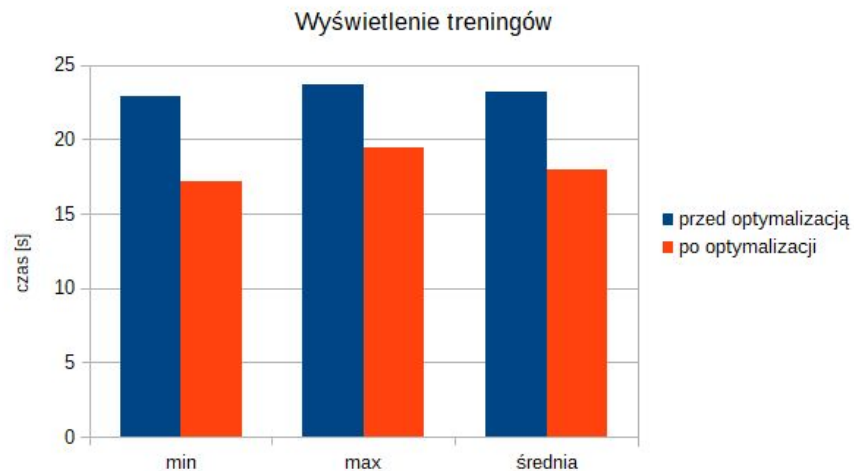
Czas [s]		
Średnia	Min	Max
11.20	10.74	12.52

Wyniki testów integracyjnych UI po optymalizacji

Tworzenie konta



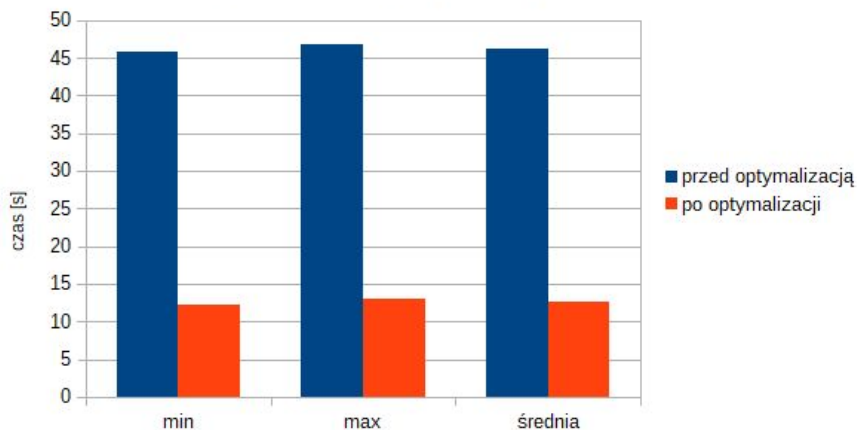
Wyświetlenie treningów



Wyniki testów integracyjnych UI po optymalizacji

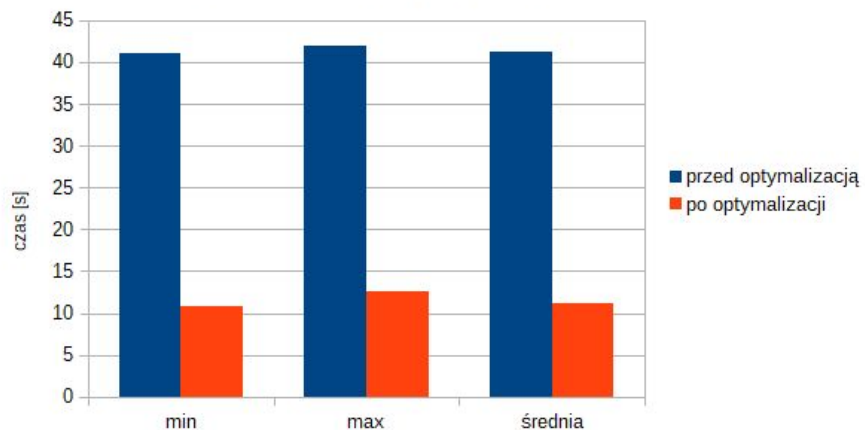
Wyświetlenie statystyk spalonych kalorii

Wyświetlenie statystyk spalonych kalorii



Wyświetlenie statystyk wykonanych ćwiczeń

Wyświetlenie statystyk wykonanych ćwiczeń



Wyniki audytu aplikacji przy użyciu narzędzia Google Lighthouse po optymalizacji

desktop

mobile



METRICS

- First Contentful Paint

0.2 s

- ▲ Total Blocking Time

900 ms

- ▲ Speed Index

2.3 s

METRICS

- First Contentful Paint

0.8 s

- ▲ Total Blocking Time

1,400 ms

- ▲ Speed Index

6.2 s

Podsumowanie wyników

Nazwa funkcji	Przewidywany czas odpowiedzi [s]	Średni czas odpowiedzi przed optymalizacją [s]	Średni czas odpowiedzi po optymalizacji [s]
Tworzenie konta	2	7,6	3,4
Logowanie na konto użytkownika	2	7,3	3,0
Utworzenie treningu	4	1,3	—
Wyświetlenie listy treningów	2	19,3	3,8
Przetwarzanie zakończenia treningu	1	3,1	2,0
Wystawienie oceny treningu	1	2,2	2,7
Czas generowania statystyk spalonych kalorii	3	3,9	2,7
Czas generowania statystyk odbytych ćwiczeń	3	2,5	2,2

Wnioski

Dodanie paginacji.

Wyniki (przy 50 aktywnych użytkownikach) uległy znacznej poprawie. Średni czas oczekiwania spadł o prawie 6s, minimalny czas odpowiedzi nieznacznie się skrócił a maksymalny czas odpowiedzi został skrócony o ponad 11s. Usunięcie wąskiego gardła nie spowodowało, że wyniki są zgodne z założeniami postawionymi przed wykonaniem testów, lecz widać ogromną poprawę.

Wnioski

Parametryzacja komendy uruchamiającej aplikację serwerową.

Sparymetryzowanie komendy uruchamiającej aplikację serwerową spowodowało obniżenie średnich, minimalnych i maksymalnych czasów odpowiedzi na zapytania dla wszystkich funkcjonalności. Zmiana została uznana za korzystną, ponieważ sprawiła znaczną poprawę wydajności aplikacji serwerowej i pozwoliła się zbliżyć do oczekiwanych założeń, postawionych przed etapem testów.

Wnioski

Zaktualizowanie Fluttera i pluginów do najnowszej wersji.

Najnowsza wersja Fluttera (3.10) usprawnia wczytywanie obrazów z API, co można zauważyć po kilkukrotnej poprawie czasów wykonania testów integracyjnych wymagających ładowania obrazów.

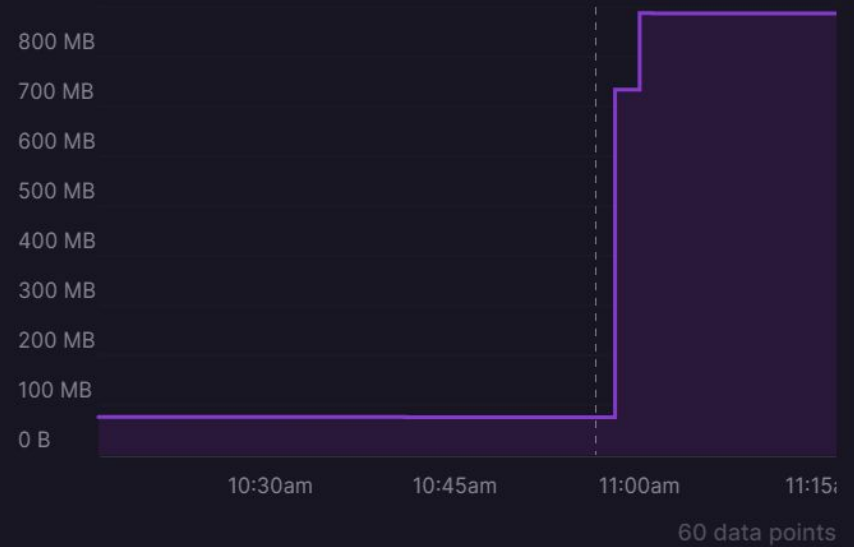
Optymalizacja poprawiła czasy FCP i Speed Index dla wersji mobilnej (aktualizacja bibliotek i Fluttera poprawiła transformację strony z wersji desktopowej w mobilną). W obydwu przypadkach, zwiększył się czas Total Blocking Time w związku z większą ilością Long Tasków (tasków wykonujących się dłużej niż 50 ms).

Optymalizacja komercyjna

CPU



Memory



Optymalizacja komercyjna

Przy obciążeniu 50 użytkowników potrzeba 0.45 CPU i 880 MB RAMu. Takie same czasy dla 100 użytkowników można osiągnąć przy zwiększeniu zasobów do ok. 1 CPU oraz prawie 2 GB RAMu. Takie same czasy dla 200 użytkowników wymagać będą 2 CPU oraz 4 GB RAMu. Stosunek liczby użytkowników do przyrostu zasobów przypomina funkcję liniową.

Na podstawie wykonanych testów oszacowano, że wraz z wzrostem liczby użytkowników, czasy odpowiedzi będą rosnać liniowo a po przekroczeniu około 250 użytkowników zaobserwowano by wzrost wykładniczy (przy aktualnej specyfikacji serwera). Odpowiedzią mogłoby być zwiększenie zasobów serwera według powyższych instrukcji.

<https://railway.app/pricing>

Dziękujemy za uwagę

Patryk Jędrzejewski

Filip Kaczmarek

Andrzej Kapczyński