Zarządzanie aplikacjami internetowymi

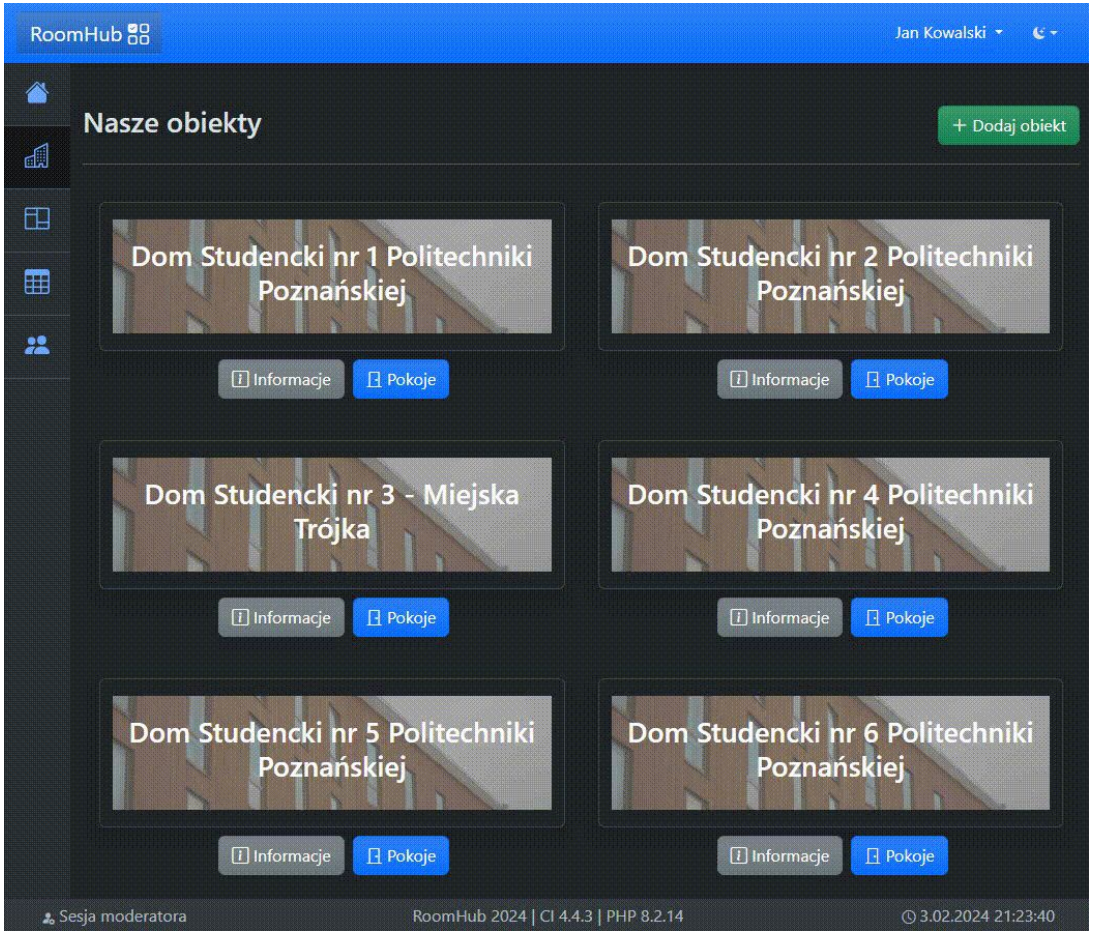
Autorzy:
Dawid Kaj, 148316
Łukasz Jankowski, 148081

Cel aplikacji

Aplikacja przeznaczona jest do zarządzania kompleksem domów studenckich.

Pracownicy mają możliwość wprowadzania i modyfikowania danych, takich jak informacje o wolnych miejscach w akademikach.

Mieszkańcy natomiast mogą przeglądać dane związane ze swoim pobytem.



Oczekiwane czasy działania poszczególnych funkcji aplikacji

Funkcja aplikacji	Czas [s]
Rejestracja użytkownika	2.0
Logowanie użytkownika	1.0
Dodawanie obiektu	1.0
Dodawanie pokoju	1.0
Dodawanie rodzaju pokoju	1.0
Dodawanie rezerwacji (wyłącznie działanie po potwierdzeniu dodania, ponieważ proces przydzielania osoby do pokoju składa się z kilku etapów)	1.5
Wyszukiwanie rezerwacji (według filtrów)	1.5
Wyświetlenie panelu mieszkańca (zawiera informacje o rezerwacjach zalogowanego użytkownika)	1.5
Usunięcie elementu (rezerwacja, rodzaj pokoju, miejsce, pokój, budynek)	1.5

Skalowalność aplikacji

Aplikacja ma za zadanie obsługiwać wielu użytkowników jednocześnie - są nimi pracownicy oraz mieszkańcy.

Oczekiwana liczba pracowników administracji korzystająca z oprogramowania w jednej chwili z aplikacji to od 5 do 10. W tym samym czasie do swoich danych mogą mieć wgląd mieszkańcy, przykładowo od 50 do 100 osób. Taki przykład mógłby odwzorować moment działania systemu w dniu roboczym, gdy rozpoczyna się lub kończy nowy miesiąc rozliczeniowy - w jednej chwili wiele osób sprawdza status uiszczenia opłaty za daną rezerwację.

Wdrożenie aplikacji

Do przetestowania wydajności aplikacji w środowisku chmurowym skorzystano z usługi EC2 w Amazon Web Services. Wybrany typ maszyny wirtualnej “t2.micro” oferuje:

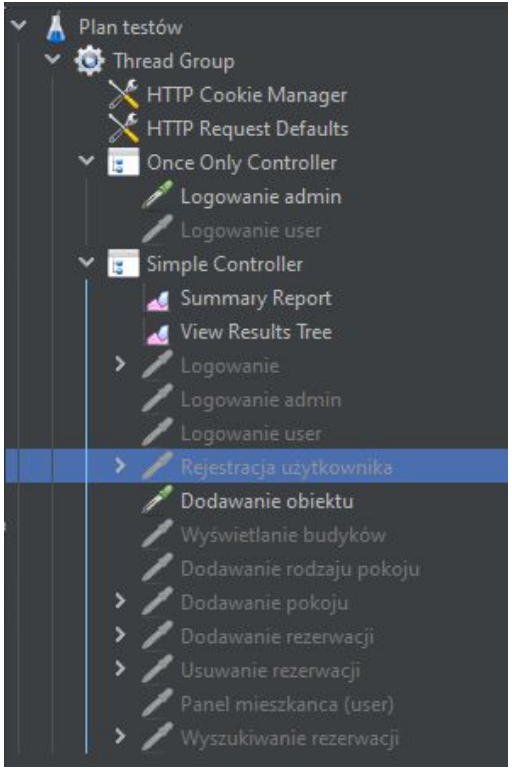
- 1 vCPU - współdzielony rdzeń wirtualny procesora z rodziny Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz (w regionie eu-west-1 - Irlandia)
- 1 GB pamięci RAM (w systemie operacyjnym dostępne 974 MB)

Na serwerze zainstalowano system operacyjny Debian 12, a następnie skonfigurowano programy dla serwera WWW (Apache w wersji 2.4.57) oraz bazy danych (MariaDB w wersji 10.11.6). Po uruchomieniu aplikacji do zarządzania kompleksem domów studenckich sprawdzono, czy wszystkie wymagane funkcje działają poprawnie.

Badanie efektywności aplikacji

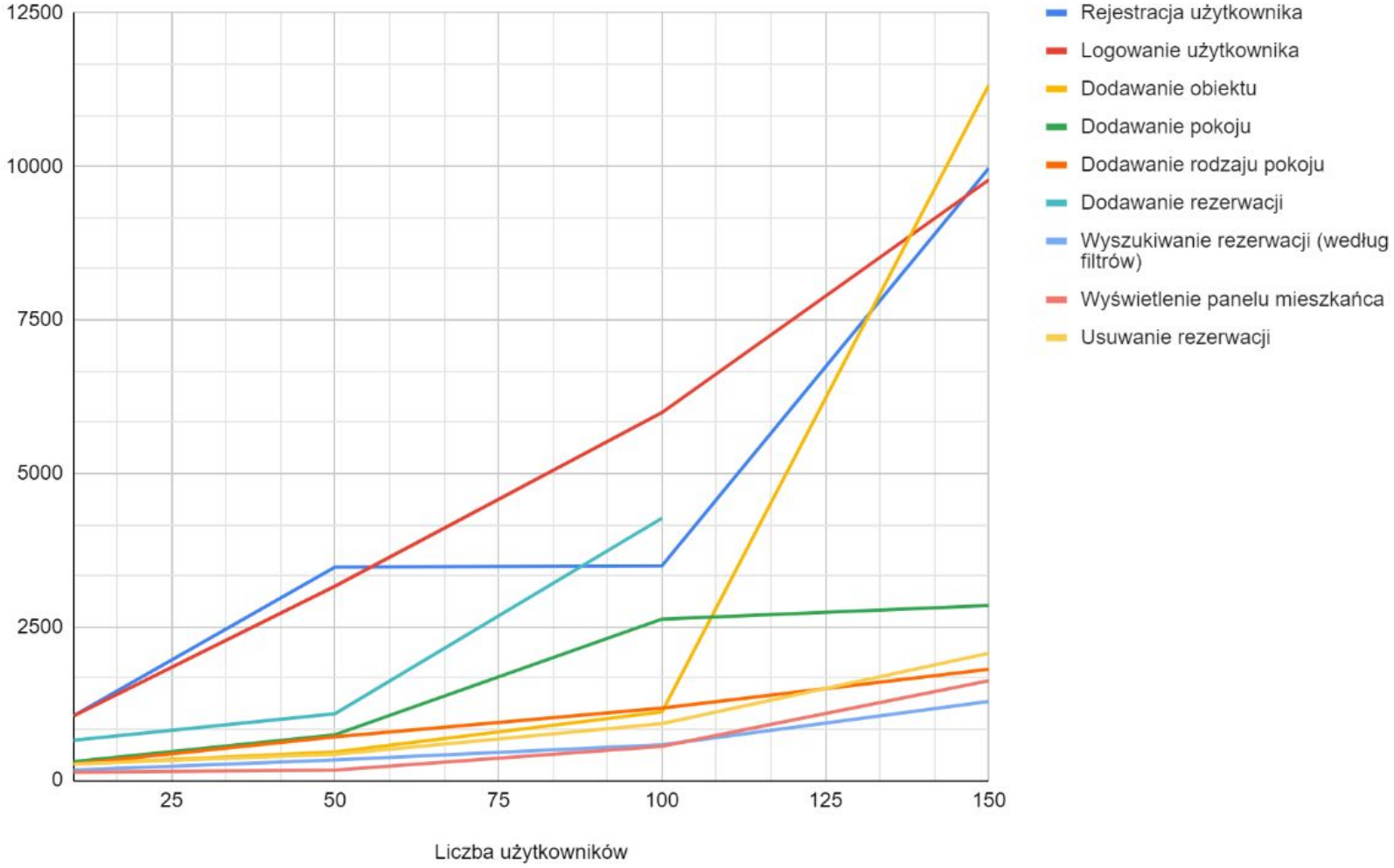
Czas działania funkcji w “ramp up”, gdy wielu użytkowników próbuje wykonać tę samą lub podobną akcję, np. wielu mieszkańców próbuje wyświetlić dane o swoich rezerwacjach.

Testy obciążenia wykonano za pomocą narzędzia Apache JMeter



Wyniki pomiarów badania efektywności aplikacji - przed optymalizacją

Średni czas załadowania strony [ms]



Podsumowanie testów aplikacji przed optymalizacją

Podczas testów monitorowano obciążenie procesora za pomocą menedżera zadań. Dla większości testów z 10 i 50 użytkownikami zużycie procesora było zmienne, ale nie przekraczało 100%.

Po zwiększeniu liczby użytkowników do 100, obciążenie procesora osiągnęło maksymalny poziom i spadało dopiero po zakończeniu testów.

W jednym z testów - dodawania rezerwacji przez 150 użytkowników - przerwano procedurę z powodu kilkuminutowego braku odpowiedzi serwera (w tym usługi połączenia SSH).

Proces optymalizacji działania aplikacji

- Ustawienia środowiskowe szkieletu programistycznego
- Optymalizacja kodu źródłowego
- Ustawienia systemu operacyjnego

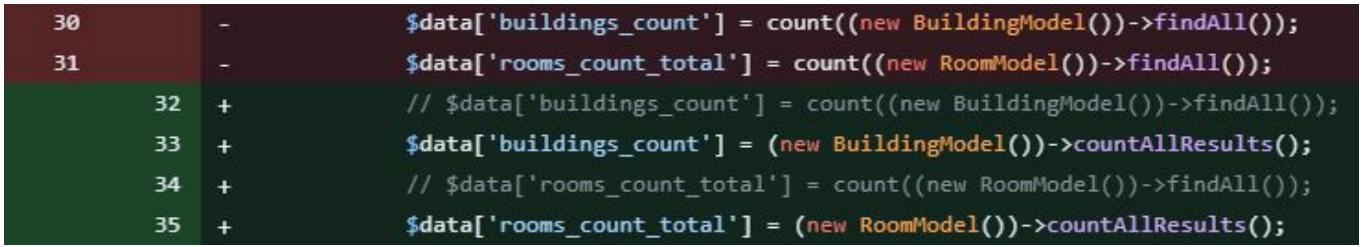
Ustawienia środowiskowe szkieletu programistycznego

Prace nad optymalizacją rozpoczęto od sprawdzenia ustawień środowiskowych CodeIgniter 4, na którym opiera się testowane oprogramowanie. W głównym katalogu znajduje się plik źródłowy “*.env*”, w którym zdefiniowane są główne zmienne aplikacji, przykładowo parametry połączenia z bazą danych i adres URL strony WWW.

Po krótkiej analizie dokumentacji odkryto, że istotna będzie zmiana ustawienia “CI_ENVIRONMENT”, które definiuje tryb działania szkieletu programistycznego. Tryb “*development*” był pomocny podczas rozwoju aplikacji - oprogramowanie zwracało szczegółowe informacje o występujących błędach. W skład tego trybu wchodziła także przydatna nakładka. Za jej pomocą można było podejrzeć na przykład treść zapytania do bazy danych. Wymienione funkcje zostały wyłączone, gdy zastosowano tryb “production”.

Optymalizacja kodu źródłowego

W ramach tej części zwrócono uwagę na te fragmenty oprogramowania, które sprawiały największe problemy, takie jak dodawanie i wyszukiwanie rezerwacji, rejestracja oraz logowanie. Optymalizacja kodu źródłowego polegała głównie na usprawnieniu zapytań SQL do bazy danych.



Ustawienia systemu operacyjnego

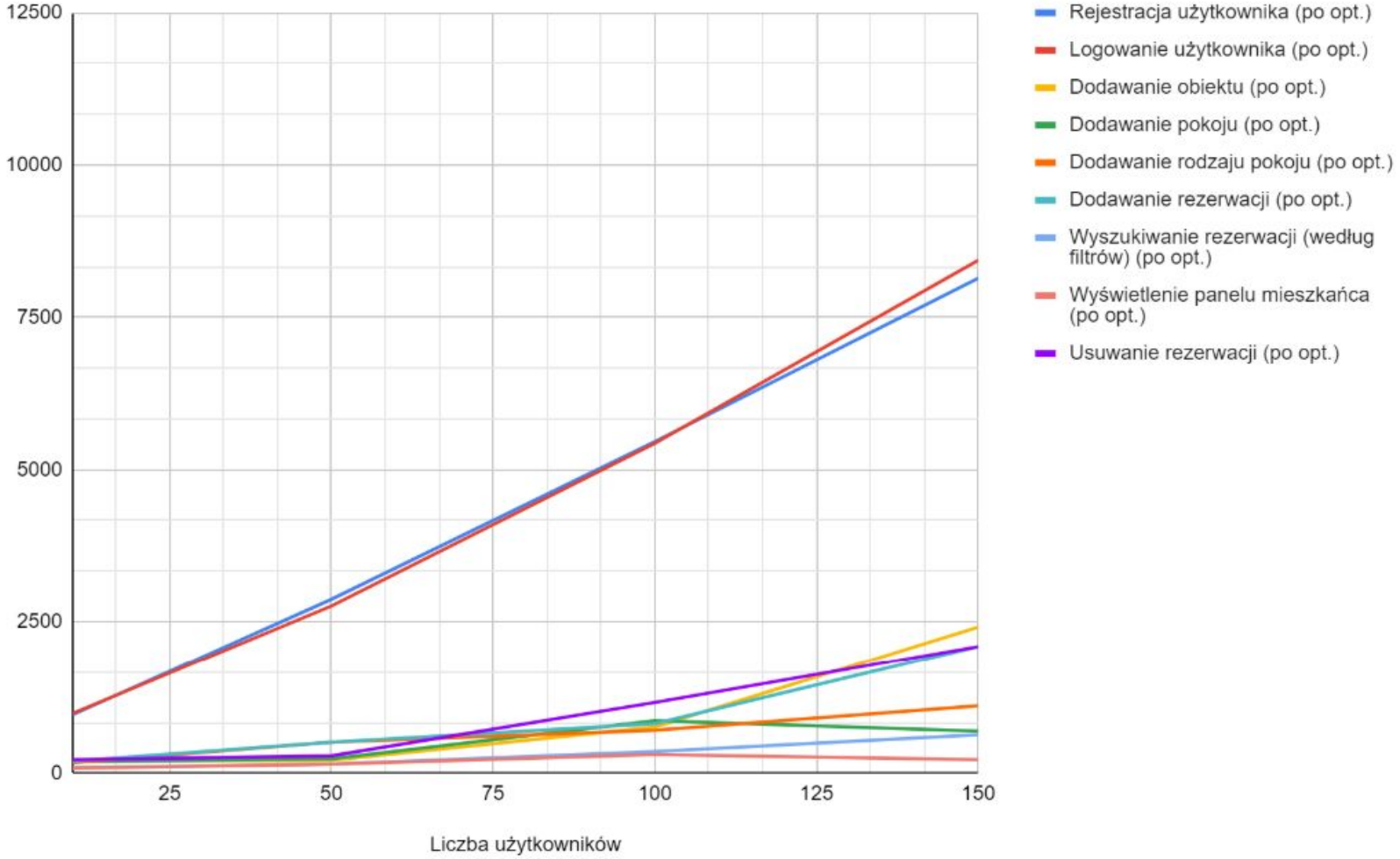
Podczas pierwszych testów aplikacji zauważono, że przy większej liczbie użytkowników może brakować pamięci RAM. Jej zajętość pod największym obciążeniem osiągała nieraz powyżej 85% (w menedżerze zadań widoczne jest 974 MB pamięci dostępnej dla systemu operacyjnego).

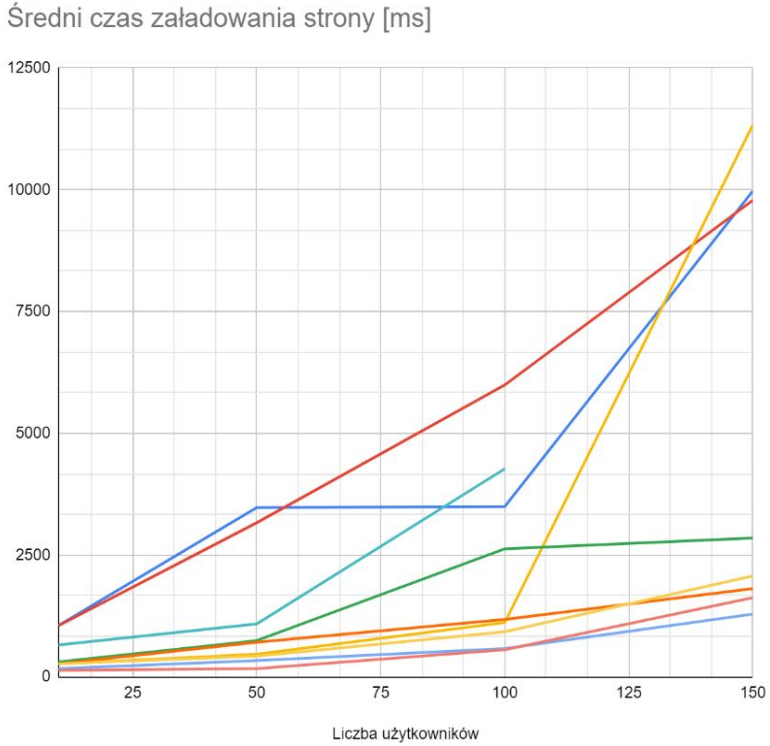
Ponieważ użyta maszyna wirtualna w AWS korzysta z dysku SSD (którego prędkość operacji odczytu i zapisu znacząco przewyższa dyski talerzowe - HDD), podjęto decyzję o utworzeniu pliku stronicowania o wielkości 1 GB - potocznie nazywanego “swap”em.

	total	used	free	shared	buff/cache	available
Mem:	976Mi	85Mi	612Mi	0.0Ki	279Mi	756Mi
Swap:	1.0Gi	0B	1.0Gi			

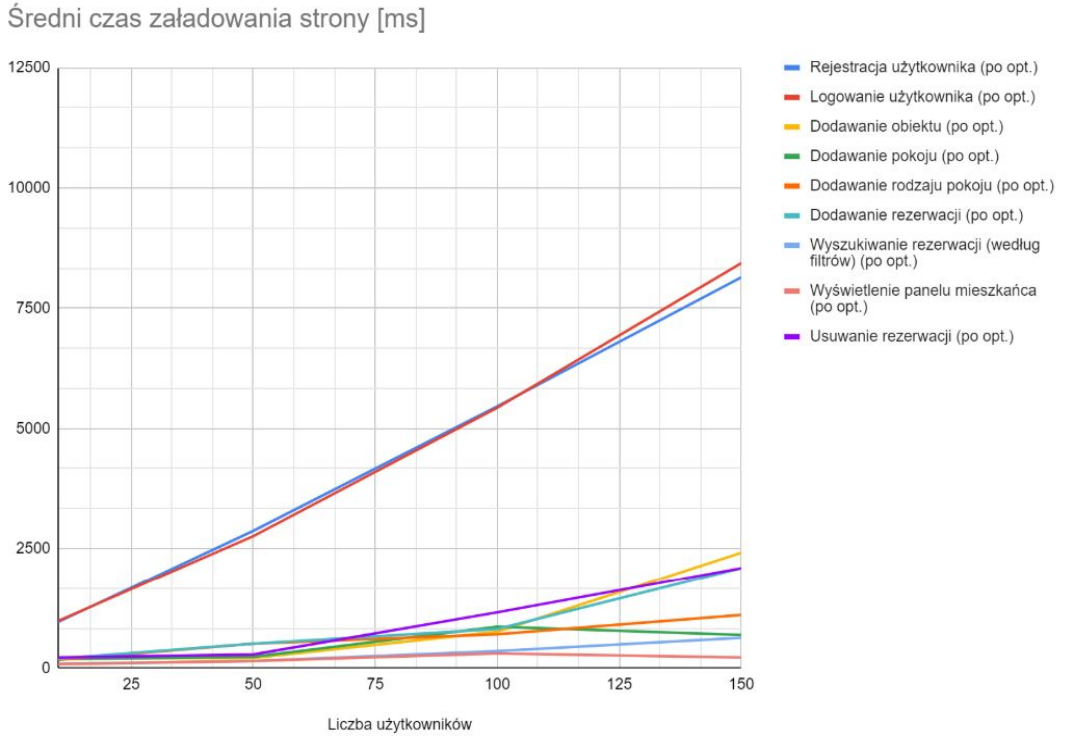
Wyniki pomiarów badania efektywności aplikacji - po optymalizacji

Średni czas załadowania strony [ms]




Średni czas załadowania strony [ms]
12500
10000
7500
5000
2500
0
25
50
75
100
125
150
Liczba użytkowników

PRZED


Średni czas załadowania strony [ms]
12500
10000
7500
5000
2500
0
25
50
75
100
125
150
Liczba użytkowników
Rejestracja użytkownika (po opt.)
Logowanie użytkownika (po opt.)
Dodawanie obiektu (po opt.)
Dodawanie pokoju (po opt.)
Dodawanie rodzaju pokoju (po opt.)
Dodawanie rezerwacji (po opt.)
Wyszukiwanie rezerwacji (według filtrów) (po opt.)
Wyświetlenie panelu mieszkańca (po opt.)
Usuwanie rezerwacji (po opt.)

PO

Podsumowanie testów aplikacji po optymalizacji

Udało się osiągnąć widoczną poprawę wyników dla tych fragmentów aplikacji, przy których w obciążeniu występowały największe problemy. Dla 150 użytkowników wciąż można zaobserwować dłuższy czas ładowania strony (szczególnie przy rejestracji i logowaniu, gdzie wciąż osiągane jest niemal 100% zajętości procesora), ale nie powtórzyła się sytuacja kilkuminutowego braku odpowiedzi - system był dostępny cały czas.

Przeniesienie części obliczeń z procesu PHP na bazę danych okazało się dobrym sposobem na poprawę wydajności zaprojektowanego oprogramowania

W szerszej perspektywie proponowaną zmianą w infrastrukturze byłoby przeniesienie bazy danych do zewnętrznej maszyny wirtualnej lub instancji bazy danych (przykładowo w usłudze AWS RDS). Mogłoby to ułatwić zarządzanie kopią zapasową oraz odciążyć procesor, który obecnie jest dzielony między proces aplikacji i proces oprogramowania bazy danych. Niestety, zwiększyłoby to koszty utrzymania aplikacji internetowej.