

Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat:	„LOKALizator” – WebAplikacja do ogłoszeń lokalnych
Autorzy:	Przemysław Niemiec, Patryk Surmacz
Grupa:	I1-210A
Kierunek:	informatyka
Rok akademicki:	2
Poziom i semestr:	I/4
Tryb studiów:	niestacjonarne

Spis treści

1. Odnośniki do innych źródeł	4
2. Słownik pojęć	5
Adaptowalność	5
Anonimowe ogłoszenie	5
Backend Developer	5
Bezpieczeństwo	5
DevOps Engineer	5
Diagram UML	5
Dostępność	5
Frontend Developer	5
Harmonogram prac	5
Jakość	5
Kamienie milowe	6
Ogłoszenie	6
Postman	6
Project Manager	6
Selenium	6
Scenariusz testowy	6
Tester	6
Test funkcjonalny	6
Użytkownik	6
UX/UI Designer	6
Wykres Gantt'a	7
Wydajność	7
Zasoby ludzkie	7
Zarządzanie jakością	7
Zarządzanie ryzykiem	7
Zarządzanie projektem	7
3. Wprowadzenie	8
3.1 Cel dokumentacji	8
3.2 Przeznaczenie dokumentacji	8
4. Specyfikacja wymagań	10
4.1 Charakterystyka ogólna	10
o Definicja produktu	10
o Podstawowe założenia	10
o Cel biznesowy	10
o Użytkownicy	10
o Korzyści z systemu	11
o Ograniczenia projektowe i wdrożeniowe	11

4.2 Wymagania funkcjonalne	11
o Lista wymagań	11
o Diagramy przypadków użycia	12
o Szczegółowy opis wymagań	15
4.3 Wymagania niefunkcjonalne	18
5. Zarządzanie projektem	20
5.1 Zasoby ludzkie	20
5.2 Harmonogram prac	20
5.3 Etapy/kamienie milowe projektu	21
6. Zarządzanie jakością	22
6.1 Scenariusze i przypadki testowe	22
7. Projekt techniczny	26
7.1 Opis architektury systemu	26
7.2 Technologie implementacji systemu	26
8.3 Diagramy UML	27
o Diagram klas	27
o Diagramy czynności	28
o Diagramy sekwencji	29
o Inne diagramy	32
7.4 Charakterystyka zastosowanych wzorców projektowych	32
1. Wzorzec MVC (Model-View-Controller)	32
2. Wzorzec Singleton	33
3. Wzorzec Dependency Injection	33
7.5 Projekt bazy danych	34
7.6 Projekt interfejsu użytkownika	35
Lista głównych elementów interfejsu	36
▪ Przejścia między głównymi elementami	37
7.7 Procedura wdrożenia	37
8. Podsumowanie	38
Szczegółowe nakłady projektowe członków zespołu	38

1. Odnosniki do innych źródeł

tj. do wykorzystywanych narzędzi / projektów w tych narzędziach:

GITHUB:

https://github.com/vinze-ps/collage_software_engineering_LOKALizator

Trello:

<https://trello.com/b/ZUx9IyGC/in%C5%BCynieria-oprogramowania-lokalizator>

2. Słownik pojęć

Adaptowalność

Możliwość dostosowania i rozbudowy systemu w odpowiedzi na zmieniające się wymagania rynku lub regulacje prawne.

Anonimowe ogłoszenie

Ogłoszenie dodane przez użytkownika bez konieczności logowania się do systemu, umożliwiające publikację treści bez ujawniania tożsamości.

Backend Developer

Osoba odpowiedzialna za projektowanie i implementację architektury serwerowej oraz baz danych, głównie w technologii Laravel 11.

Bezpieczeństwo

Mechanizmy i procedury mające na celu ochronę danych przed nieautoryzowanym dostępem, utratą lub zniszczeniem.

DevOps Engineer

Osoba odpowiedzialna za konfigurację środowisk, zarządzanie wersjami oraz wdrożenie ostatecznego produktu.

Diagram UML

Graf przedstawiający różne aspekty systemu, takie jak struktura, zachowanie czy interakcje, w celu ułatwienia jego projektowania i zrozumienia.

Dostępność

Stopień, w jakim system jest dostępny do użytku, mierzone jako procent czasu, w którym system działa prawidłowo.

Frontend Developer

Osoba odpowiedzialna za projektowanie i implementację interfejsu użytkownika oraz interakcji z API, korzystając z biblioteki React.

Harmonogram prac

Plan określający etapy realizacji projektu, czas trwania poszczególnych zadań i kamieni milowych, często przedstawiany za pomocą wykresu Gantta.

Jakość

Stopień spełnienia przez system wymagań funkcjonalnych i нефункциональных, w tym wydajności, bezpieczeństwa, użyteczności i innych cech jakości.

Kamienie milowe

Kluczowe etapy w projekcie, które oznaczają osiągnięcie ważnych celów i postępów.

Ogłoszenie

Treść publikowana przez użytkownika w systemie, zawierająca informacje o przedmiocie lub usłudze do sprzedaży, wynajmu lub wymiany.

Postman

Narzędzie wspomagające testowanie API, umożliwiające wysyłanie zapytań HTTP i analizowanie odpowiedzi.

Project Manager

Osoba odpowiedzialna za nadzorowanie wszystkich etapów projektu, zarządzanie zespołem i komunikację z interesariuszami.

Selenium

Narzędzie wspomagające automatyzację testów przeglądarkowych, umożliwiające symulowanie działań użytkowników i testowanie interfejsów webowych.

Scenariusz testowy

Opis kroków, które należy wykonać, aby przetestować określoną funkcjonalność systemu, w tym dane wejściowe, działania testera i oczekiwane wyniki.

Tester

Specjalista odpowiedzialny za testowanie systemu w celu zapewnienia jego jakości i spełnienia wymagań funkcjonalnych oraz нефункциональных.

Test funkcjonalny

Rodzaj testu mającego na celu sprawdzenie, czy system działa zgodnie z wymaganiami funkcjonalnymi.

Użytkownik

Osoba korzystająca z systemu, w tym indywidualni sprzedawcy, kupujący oraz firmy lokalne.

UX/UI Designer

Osoba odpowiedzialna za projektowanie estetyczne i funkcjonalne interfejsu użytkownika, zapewniając przyjazne doświadczenie dla użytkownika końcowego.

Wykres Gantta

Graficzna reprezentacja harmonogramu projektu, pokazująca czas trwania zadań i etapy realizacji.

Wydajność

Zdolność systemu do szybkiego i efektywnego przetwarzania danych oraz obsługi wielu użytkowników jednocześnie.

Zasoby ludzkie

Osoby zaangażowane w realizację projektu, w tym developerzy, testerzy, project manager, UX/UI designer oraz DevOps engineer.

Zarządzanie jakością

Proces zapewniający, że system spełnia określone standardy jakości poprzez testowanie i walidację funkcji.

Zarządzanie ryzykiem

Proces identyfikacji, oceny i zarządzania ryzykami, które mogą wpływać na projekt, w celu minimalizacji ich negatywnego wpływu.

Zarządzanie projektem

Proces planowania, organizowania i nadzorowania wszystkich aspektów projektu, w tym zarządzania zasobami, harmonogramem i budżetem.

3. Wprowadzenie

3.1 Cel dokumentacji

Celem dokumentacji projektu "Lokalizator" jest szczegółowe przedstawienie wszystkich aspektów związanych z tworzeniem, wdrażaniem oraz utrzymaniem systemu internetowego do publikacji i przeglądania lokalnych ogłoszeń. Dokumentacja ta ma na celu zapewnienie kompleksowego wglądu w proces projektowania i implementacji aplikacji, ułatwienie komunikacji między członkami zespołu oraz interesariuszami, a także zapewnienie jasnych wytycznych i standardów dla przyszłych działań związanych z rozwojem i wsparciem systemu.

Główne cele dokumentacji obejmują:

- **Opis systemu i jego funkcjonalności:** Dostarczenie szczegółowych informacji na temat działania systemu, w tym specyfikacji technicznych, wymagań funkcjonalnych i niefunkcjonalnych.
- **Zarządzanie projektem:** Ułatwienie planowania, monitorowania i kontrolowania wszystkich etapów projektu, w tym zarządzania zasobami, harmonogramem i ryzykiem.
- **Zarządzanie jakością:** Zapewnienie, że system spełnia określone standardy jakości poprzez dokładne testowanie i walidację wszystkich funkcji.
- **Wsparcie dla użytkowników końcowych:** Dostarczenie szczegółowych instrukcji i informacji, które pomogą użytkownikom końcowym w efektywnym korzystaniu z systemu.

3.2 Przeznaczenie dokumentacji

Przeznaczeniem dokumentacji projektu "Lokalizator" jest służyć jako główne źródło informacji dla różnych grup interesariuszy zaangażowanych w rozwój, wdrożenie i użytkowanie systemu. Dokumentacja jest skierowana do następujących odbiorców:

- **Członkowie zespołu projektowego:** Developerzy, projektanci, testerzy i managerowie projektu, którzy potrzebują szczegółowych informacji technicznych oraz harmonogramu prac, aby efektywnie realizować swoje zadania.
- **Zarząd i interesariusze:** Osoby decyzyjne w organizacji, które potrzebują zrozumieć cel i korzyści płynące z wdrożenia systemu, a także monitorować postępy projektu.
- **Testerzy:** Specjaliści ds. jakości, którzy potrzebują scenariuszy testowych i kryteriów akceptacji, aby przeprowadzać dokładne testy funkcjonalności i niefunkcjonalności systemu.
- **Użytkownicy końcowi:** Osoby korzystające z systemu, które potrzebują jasnych instrukcji dotyczących obsługi aplikacji, w tym dodawania, wyszukiwania i zarządzania ogłoszeniami.

Dokumentacja ma na celu zaspokojenie potrzeb informacyjnych wszystkich tych grup, zapewniając im dostęp do niezbędnych informacji, które pozwolą na skuteczne wdrożenie

i eksploatację systemu "Lokalizator". Jest ona żywym dokumentem, który będzie aktualizowany na różnych etapach projektu, aby odzwierciedlać zmiany w wymaganiach, specyfikacjach technicznych i innych istotnych aspektach.

4. Specyfikacja wymagań

4.1 Charakterystyka ogólna

- **Definicja produktu**

"Lokalizator" – system internetowy do publikacji i przeglądania lokalnych ogłoszeń.

- **Podstawowe założenia**

Serwis "Lokalizator" został zaprojektowany, aby służyć jako centralne miejsce dla mieszkańców danej lokalizacji do publikowania i przeglądania ogłoszeń lokalnych. Serwis wzorowany jest na facebookowych stronach typu „Spotted”. Będzie więc zawierał zarówno „luźne” ogłoszenia, pozdrowienia czy pytania, jak również oferty kupna, sprzedaży czy usług. Platforma ma również na celu integrację lokalnej społeczności i promocję lokalnej gospodarki poprzez umożliwienie bezpośredniego kontaktu między mieszkańcami. Ogłoszenia będzie można publikować zarówno anonimowo (na wzór „Spotted”) jak i pod własnym nazwiskiem.

- **Cel biznesowy**

Celem biznesowym "Lokalizatora" jest stworzenie stabilnej i dochodowej platformy ogłoszeniowej, która zdobędzie popularność i zaufanie w lokalnych społecznościach. System ma na celu przyciągnięcie dużej liczby użytkowników dzięki swojej użyteczności, prostocie i dostępności, co w konsekwencji ma przekładać się na zyski z reklam oraz potencjalnych usług premium.

- **Użytkownicy**

- Użytkownicy anonimowi
- Indywidualni Sprzedawcy: Osoby prywatne chcące sprzedać lub wynająć swoje rzeczy lub usługi.
- Kupujący: Osoby szukające produktów lub usług do zakupu lub wynajmu.
- Firmy lokalne: Przedsiębiorstwa zainteresowane reklamowaniem swoich produktów i usług lokalnie

- **Korzyści z systemu**

- Anonimowi użytkownicy (ID-00): Łatwość i szybkość dodawania ogłoszeń
- Indywidualni sprzedawcy (ID-01): Dostęp do klientów znajdujących się w tym samym mieście, co zwiększa szanse na szybką sprzedaż.

- Kupujący (ID-02): Dostęp do szerokiej gamy lokalnych ofert w jednym miejscu, co ułatwia porównywanie i wybieranie najlepszych opcji.
- Firmy lokalne (ID-03): Możliwość skutecznej reklamy w lokalnym środowisku, co zwiększa rozpoznawalność marki i przyciąga nowych klientów.

- **Ograniczenia projektowe i wdrożeniowe**

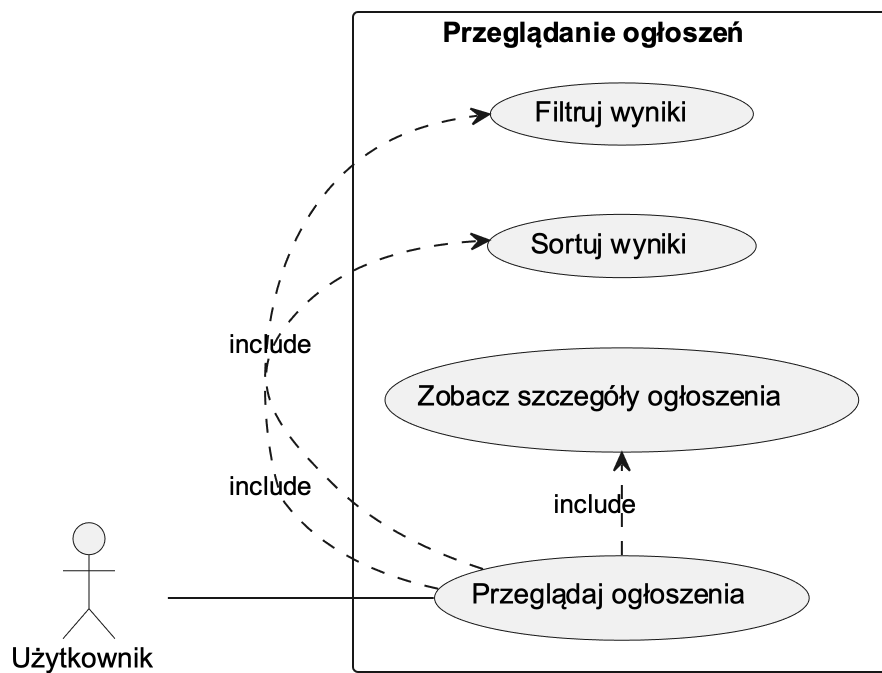
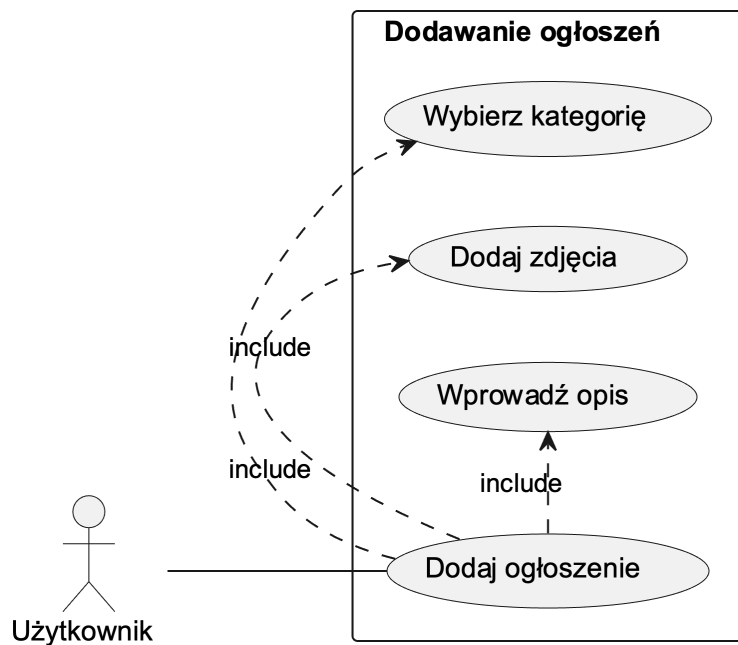
Projekt musi być zgodny z RODO w zakresie ochrony danych osobowych, co wymaga implementacji odpowiednich mechanizmów bezpieczeństwa i zasad prywatności. System musi być również dostosowany do obsługi na różnych platformach (desktop, mobile), co wymaga responsywnego projektowania interfejsów. Wymagane jest także zapewnienie wysokiej dostępności i odporności na awarie, co będzie wymagać zastosowania odpowiednich rozwiązań w infrastrukturze serwerowej.

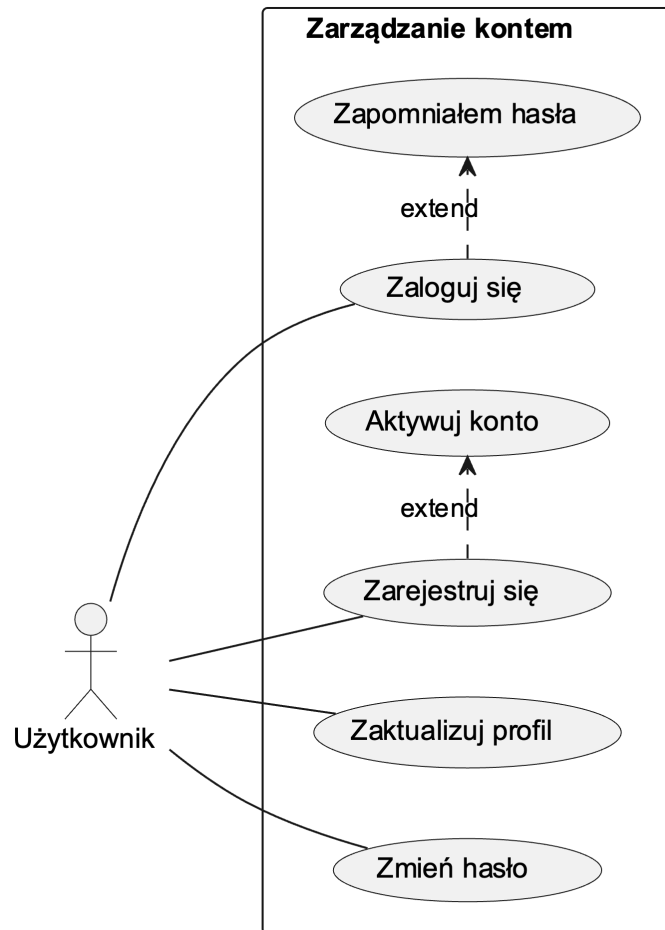
4.2 Wymagania funkcjonalne

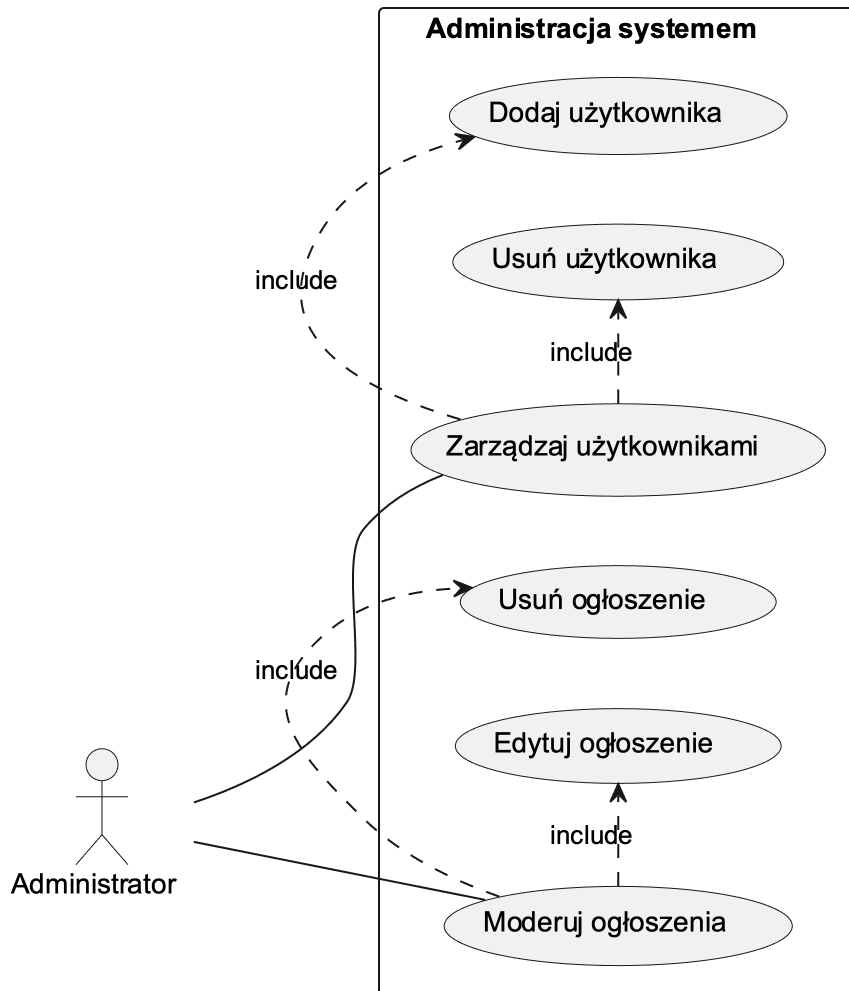
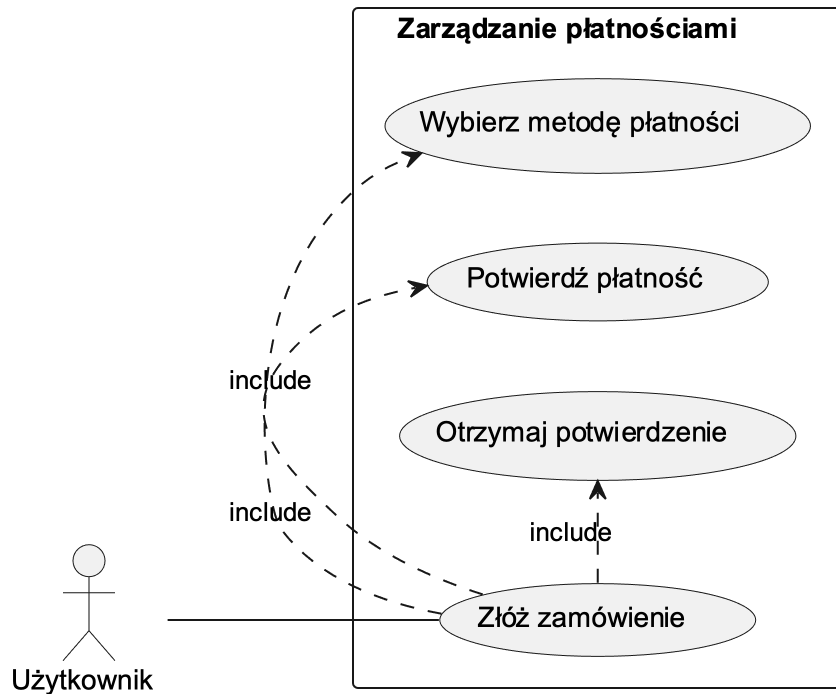
- **Lista wymagań**

- a. Użytkownik może stworzyć konto i zarządzać nim.
- b. Użytkownik może dodawać, edytować i usuwać ogłoszenia.
- c. Użytkownik może przeglądać ogłoszenia według kategorii.
- d. Użytkownik może wyszukiwać ogłoszenia używając filtrów (np. lokalizacja, rodzaj ogłoszenia, cena).
- e. Użytkownik może dodawać ogłoszenia do ulubionych.
- f. System oferuje opcje kontaktu między kupującymi a sprzedającymi.
- g. System umożliwia zarządzanie płatnościami w przypadku usług premium.

○ Diagramy przypadków użycia







- **Szczegółowy opis wymagań**

Przypadek użycia 1: Dodawanie Ogłoszenia

ID: UC-01

Nazwa: Dodawanie Ogłoszenia

Uzasadnienie biznesowe: (ID-01) Pozwala sprzedawcom na szybkie i łatwe publikowanie ofert, co jest główną domeną serwisu.

Użytkownicy: Indywidualni Sprzedawcy, Firmy Lokalne

Scenariusze:

SCENARIUSZ 1

Nazwa scenariusza: Dodawanie Standardowego Ogłoszenia

Warunki początkowe: Użytkownik jest zalogowany.

Przebieg działań:

1. Użytkownik wybiera opcję „Dodaj Ogłoszenie”.
2. Wypełnia formularz ogłoszenia (tytuł, opis, cena, zdjęcia).
3. Wybiera kategorie ogłoszenia.
4. Przesyła formularz.

Efekty: Ogłoszenie jest dodane do systemu i widoczne dla innych użytkowników.

Wymagania нефunkcjonalne: System powinien obsłużyć żądanie dodania pojedynczego ogłoszeni w czasie poniżej 3 sek.

Częstotliwość: 3 (średnio często)

Istotność: 5 (krytyczne)

SCENARIUSZ 2

Nazwa scenariusza: Dodawanie Anonimowego Ogłoszenia

Warunki początkowe: Użytkownik nie musi być zalogowany.

Przebieg działań:

1. Użytkownik wybiera opcję „Dodaj Ogłoszenie Anonimowo”.
2. Wypełnia formularz ogłoszenia bez podawania swoich danych.
3. Wybiera kategorie ogłoszenia.
4. Przesyła formularz.

Efekty: Ogłoszenie jest dodane anonimowo do systemu i widoczne dla innych użytkowników.

Wymagania нефunkcjonalne: System musi zapewnić anonimowość użytkownika.

Częstotliwość: 3 (średnio często)

Istotność: 4 (ważne)

Przypadek Użycia 2: Wyszukiwanie Ogłoszeń

ID: UC-02

Nazwa: Wyszukiwanie Ogłoszeń

Uzasadnienie biznesowe: (ID-02) Umożliwia użytkownikom efektywne odnajdywanie ofert spełniających ich kryteria, co poprawia doświadczenia użytkownika i zwiększa zaangażowanie.

Użytkownicy: Kupujący, Przeglądający

Scenariusze:

SCENARIUSZ 1

Nazwa scenariusza: Wyszukiwanie zaawansowane

Warunki początkowe: Użytkownik jest na stronie wyszukiwania.

Przebieg działań:

1. Użytkownik wpisuje słowa kluczowe w pole wyszukiwania.
2. Ustawia filtry (cena, lokalizacja, kategoria).
3. Uruchamia wyszukiwanie.
4. Przegląda wyniki.

Efekty: Wyświetlane są ogłoszenia spełniające kryteria wyszukiwania.

Wymagania нефunkcjonalne: System musi obsługiwać wyszukiwanie w czasie rzeczywistym. Czas odpowiedzi wyszukiwarki powinien wynosić poniżej 2 sek.

Częstotliwość: 5 (bardzo często)

Istotność: 5 (krytyczne)

SCENARIUSZ 2

Nazwa scenariusza: Przeglądanie ogłoszeń na stronie głównej

Warunki początkowe: Użytkownik jest na stronie głównej.

Przebieg działań:

1. Użytkownik przegląda ogłoszenia na stronie głównej
2. Użytkownik może zmienić stronę (paginacja)
3. Użytkownik może kliknąć w dowolne ogłoszenie i przeczytać o nim więcej.

Efekty: Wyświetlane są wszystkie ogłoszenia posortowane po dacie dodania malejąco.

Wymagania нефunkcjonalne: System musi wyświetlić najnowsze ogłoszenia w czasie poniżej 0.5 sek od wejścia na stronę.

Częstotliwość: 5 (bardzo często)

Istotność: 5 (krytyczne)

Przypadek Użycia 3: Zarządzanie Kontem Użytkownika

ID: UC-03

Nazwa: Zarządzanie Kontem Użytkownika

Uzasadnienie biznesowe: (ID-01, ID-03) Umożliwia użytkownikom personalizację i kontrolę nad ich kontem i ogłoszeniami.

Użytkownicy: Indywidualni Sprzedawcy, Kupujący

Scenariusze:

SCENARIUSZ 1

Nazwa scenariusza: Edycja Danych Konta

Warunki początkowe: Użytkownik jest zalogowany i jest na podstronie swojego profilu.

Przebieg działań:

1. Użytkownik wybiera „Edytuj Profil”.
2. Aktualizuje dane (np. nazwisko, adres e-mail).
3. Zatwierdza zmiany.

Efekty: Dane konta są aktualizowane.

Wymagania niefunkcjonalne: Zapewnienie bezpieczeństwa danych.

Częstotliwość: 2 (rzadko)

Istotność: 4 (ważne)

4.3 Wymagania niefunkcjonalne

1. Wydajność

Wydajność systemu "Lokalizator" musi być dostosowana do obsługi wielu użytkowników jednocześnie, szczególnie w godzinach szczytu, co jest krytyczne dla użytkowników przeglądających i dodających ogłoszenia.

Specyfikacja:

- System powinien obsługiwać do 10,000 równoczesnych użytkowników bez znaczącego spadku wydajności.
- Czas odpowiedzi systemu na żądania użytkowników nie powinien przekraczać 2 sekund.

2. Bezpieczeństwo

Bezpieczeństwo jest kluczowym aspektem, mając na uwadze ochronę danych osobowych użytkowników oraz anonimowość w przypadku dodawania ogłoszeń.

Specyfikacja:

- Wdrożenie zabezpieczeń przed atakami typu SQL injection, XSS, i CSRF.
- Stosowanie szyfrowania danych wrażliwych, w tym danych osobowych i komunikacji między klientem a serwerem.
- Regularne kopie zapasowe danych oraz strategie szybkiego przywracania systemu po ewentualnych awariach.

3. Zabezpieczenia

Zabezpieczenia obejmują zarówno środki techniczne, jak i odpowiednie procedury.

Specyfikacja:

- Implementacja certyfikatów SSL/TLS dla całego ruchu sieciowego.
- Mechanizmy autentykacji dwuetapowej dla użytkowników.
- Regularne audyty bezpieczeństwa przeprowadzane przez zewnętrzne firmy.

4. Inne cechy jakości

- **Adaptowalność:** System musi być łatwy do dostosowania i rozbudowy w odpowiedzi na zmieniające się wymagania rynku lub regulacje prawne. Każda nowa funkcja powinna być w stanie zostać wdrożona w ciągu maksymalnie 1 miesiąca.
- **Dostępność:** System powinien być dostępny 24/7, z maksymalnym czasem przestoju nie przekraczającym 99,9% dostępności rocznie.
- **Poprawność:** Wszystkie funkcje systemu muszą działać zgodnie ze swoimi specyfikacjami, bez błędów krytycznych.
- **Elastyczność:** System powinien być zdolny do skalowania w zależności od wzrostu liczby użytkowników i obciążenia.
- **Łatwość konserwacji:** Kod źródłowy powinien być napisany zgodnie ze standardami branżowymi, co ułatwia jego utrzymanie i aktualizacje.
- **Przenośność:** System powinien być kompatybilny z różnymi systemami operacyjnymi i urządzeniami (desktop, mobile).
- **Awaryjność:** System powinien posiadać mechanizmy szybkiego wykrywania błędów i ich naprawy, z minimalnym wpływem na działanie systemu.
- **Testowalność:** Architektura systemu powinna wspierać łatwe tworzenie i wykonanie testów automatycznych.
- **Użyteczność:** Interfejs użytkownika powinien być intuicyjny i prosty w obsłudze, z minimalnym czasem nauki dla nowych użytkowników.

5. Zarządzanie projektem

5.1 Zasoby ludzkie

Zespół odpowiedzialny za realizację projektu składa się z następujących ról:

- Project Manager: odpowiedzialny za nadzorowanie wszystkich etapów projektu, zarządzanie zespołem i komunikację z interesariuszami. (1 osoba)
- Backend Developer(s): odpowiedzialni za projektowanie i implementację architektury serwerowej oraz baz danych, głównie w technologii Laravel 11. (2 osoby)
- Frontend Developer(s): odpowiedzialni za projektowanie i implementację interfejsu użytkownika oraz interakcji z API, korzystając z biblioteki React. (3 osoby)
- DevOps Engineer: odpowiedzialny za konfigurację środowisk, zarządzanie wersjami oraz wdrożenie ostateczne produktu. (1 osoba)
- Quality Assurance Specialist: odpowiedzialny za testowanie całego systemu, zapewnienie jakości i spełnienie wymagań funkcjonalnych i niefunkcjonalnych. (1 osoba)
- UX/UI Designer: odpowiedzialny za projektowanie estetyczne i funkcjonalne interfejsu użytkownika, zapewniając przyjazne doświadczenie dla użytkownika końcowego. (1 osoba)

5.2 Harmonogram prac

- Faza 1: Planowanie i Projektowanie (1 miesiąc)
 - Analiza wymagań
 - Projekt architektury systemu i bazy danych
 - Projekt interfejsu użytkownika
- Faza 2: Implementacja (2 miesiące)
 - Budowa backendu
 - Budowa frontendu
 - Integracja frontendu z backendem
- Faza 3: Testowanie (1 miesiąc)
 - Testy jednostkowe i integracyjne
 - Testy użytkowe
 - Rozwiązywanie wykrytych błędów
- Faza 4: Wdrożenie (2 tygodnie)
 - Konfiguracja środowiska produkcyjnego
 - Wdrożenie ostateczne
 - Szkolenie użytkowników

5.3 Etapy/kamienie milowe projektu

Główne kamienie milowe projektu to:

- Zakończenie analizy wymagań i zatwierdzenie projektu (koniec Fazy 1)
- Ukończenie głównych modułów backendu i frontendu (koniec Fazy 2)
- Zakończenie wszystkich testów i potwierdzenie jakości (koniec Fazy 3)
- Pomyślne wdrożenie i akceptacja systemu przez użytkowników (koniec Fazy 4)

Każdy z tych kamieni milowych będzie monitorowany przez Project Managera, aby zapewnić, że projekt przebiega zgodnie z planem i osiąga wszystkie wyznaczone cele.

6. Zarządzanie jakością

6.1 Scenariusze i przypadki testowe

Test Case 1: Dodawanie Nowego Ogłoszenia

- **ID:** TC-01
- **Nazwa scenariusza:** Dodawanie Nowego Ogłoszenia
- **Kategoria:** Test funkcjonalny
- **Opis:** Sprawdzenie, czy użytkownik może poprawnie dodać nowe ogłoszenie.
- **Tester:** Jan Kowalski
- **Termin:** 2024-06-01
- **Narzędzia wspomagające:** Postman, Selenium
- **Założenia, środowisko, warunki wstępne, dane wejściowe:**
 - Użytkownik musi być zalogowany.
 - Przeglądarka internetowa z dostępem do serwera testowego.

Przebieg działań

Lp.	Działanie testera	Oczekiwane działanie systemu
1	Użytkownik loguje się na swoje konto.	System autoryzuje użytkownika i przekierowuje do strony głównej.
2	Użytkownik wybiera opcję „Dodaj Ogłoszenie”.	System wyświetla formularz dodawania ogłoszenia.
3	Użytkownik wypełnia formularz (tytuł, opis, cena, zdjęcia).	System przyjmuje dane i weryfikuje poprawność pól.
4	Użytkownik klika przycisk „Zapisz”.	System zapisuje ogłoszenie i wyświetla potwierdzenie.

Zestaw danych testowych

Lp.	Dane wejściowe	Oczekiwane dane wyjściowe
1	Tytuł: "Sprzedam rower", Opis: "Używany, w dobrym stanie", Cena: 500, Zdjęcia: "rower.jpg"	Komunikat: "Ogłoszenie zostało dodane pomyślnie"
2	Tytuł: "Wykonam renowację starych płyt CD", Opis: "", Cena: 30, Zdjęcia: "plyta.png"	Komunikat: "Ogłoszenie zostało dodane pomyślnie"

Warunek zaliczenia testu: Ogłoszenie zostaje dodane i jest widoczne w systemie.

Test Case 2: Wyszukiwanie Ogłoszenia

- **ID:** TC-02
- **Nazwa scenariusza:** Wyszukiwanie Ogłoszenia
- **Kategoria:** Test funkcjonalny
- **Opis:** Sprawdzenie, czy użytkownik może wyszukać ogłoszenia według określonych kryteriów.
- **Tester:** Anna Nowak
- **Termin:** 2024-06-02
- **Narzędzia wspomagające:** Postman, Selenium
- **Założenia, środowisko, warunki wstępne, dane wejściowe:**
 - W systemie istnieją ogłoszenia spełniające kryteria wyszukiwania.
 - Przeglądarka internetowa z dostępem do serwera testowego.

Przebieg działań

Lp.	Działania testera	Oczekiwane działania systemu
1	Użytkownik przechodzi do strony wyszukiwania.	System wyświetla formularz wyszukiwania.
2	Użytkownik wpisuje kryteria wyszukiwania (np. "rower", cena do 500).	System przetwarza kryteria i wykonuje zapytanie.
3	Użytkownik klika przycisk "Szukaj".	System wyświetla listę ogłoszeń spełniających kryteria.

Zestaw danych testowych

Lp.	Dane wejściowe	Oczekiwane dane wyjściowe
1	Fraza: "rower", Cena maks: 500	Lista ogłoszeń zawierających "rower" w cenie do 500

Warunek zaliczenia testu: System wyświetla prawidłowe ogłoszenia spełniające podane kryteria wyszukiwania.

Test Case 3: Edycja Ogłoszenia

- **ID:** TC-03
- **Nazwa scenariusza:** Edycja Ogłoszenia
- **Kategoria:** Test funkcjonalny
- **Opis:** Sprawdzenie, czy użytkownik może edytować istniejące ogłoszenie.
- **Tester:** Piotr Wójcik
- **Termin:** 2024-06-03
- **Narzędzia wspomagające:** Postman, Selenium
- **Założenia, środowisko, warunki wstępne, dane wejściowe:**
 - Użytkownik jest zalogowany.
 - Ogłoszenie istnieje w systemie i należy do użytkownika.

Przebieg działań

Lp.	Działania testera	Oczekiwane działania systemu
1	Użytkownik loguje się.	System autoryzuje użytkownika.
2	Użytkownik wybiera ogłoszenie do edycji.	System wyświetla szczegóły ogłoszenia.
3	Użytkownik zmienia dane (np. opis, cena).	System przyjmuje nowe dane i weryfikuje poprawność.
4	Użytkownik zapisuje zmiany.	System aktualizuje ogłoszenie.

Zestaw danych testowych

Lp.	Dane wejściowe	Oczekiwane dane wyjściowe
1	Opis: "Używany, w bardzo dobrym stanie", Cena: 450	Komunikat: "Zmiany zostały zapisane"

Warunek zaliczenia testu: Ogłoszenie jest aktualizowane i zmiany są widoczne w systemie.

Test Case 4: Usuwanie Ogłoszenia

- **ID:** TC-04
- **Nazwa scenariusza:** Usuwanie Ogłoszenia
- **Kategoria:** Test funkcjonalny
- **Opis:** Sprawdzenie, czy użytkownik może usunąć swoje ogłoszenie.
- **Tester:** Karolina Lewandowska
- **Termin:** 2024-06-04
- **Narzędzia wspomagające:** Postman, Selenium
- **Założenia, środowisko, warunki wstępne, dane wejściowe:**
 - Użytkownik jest zalogowany.
 - Ogłoszenie istnieje w systemie i należy do użytkownika.

Przebieg działań

Lp.	Działania testera	Oczekiwane działania systemu
1	Użytkownik loguje się.	System autoryzuje użytkownika.
2	Użytkownik wybiera ogłoszenie do usunięcia.	System wyświetla szczegóły ogłoszenia.
3	Użytkownik klika "Usuń".	System wyświetla potwierdzenie.
4	Użytkownik potwierdza usunięcie.	System usuwa ogłoszenie i wyświetla komunikat.

Zestaw danych testowych

Lp.	Dane wejściowe	Oczekiwane dane wyjściowe
1	Ogłoszenie ID: 123	Komunikat: "Ogłoszenie zostało usunięte"

Warunek zaliczenia testu: Ogłoszenie jest usunięte i nie jest widoczne w systemie.

7. Projekt techniczny

7.1 Opis architektury systemu

"LOKALizator" jest oparty na architekturze opartej na modelu klient-serwer. Serwer API stworzony jest w Laravel 11, zapewniając wydajne zarządzanie danymi i logiką biznesową, natomiast klient (frontend) zbudowany jest z użyciem ReactJS.

- **Frontend:** Aplikacja webowa stworzona w React.js, zapewniająca dynamiczne i responsywne interfejsy użytkownika, które komunikują się z backendem za pośrednictwem RESTful API.
- **Backend:** Aplikacja serwerowa w Laravel 11, która obsługuje wszystkie żądania z frontendu, zarządzanie danymi, autentykację użytkowników i inne procesy biznesowe.
- **Baza danych:** PostgreSQL, wybrany dla swojej niezawodności i wsparcia dla zaawansowanych funkcji SQL, który jest odpowiedni do zarządzania relacyjnymi danymi w aplikacjach o dużym obciążeniu.

7.2 Technologie implementacji systemu

Technologia	Uzasadnienie wyboru
Laravel 11	Laravel to popularny framework PHP, który zapewnia bogaty zestaw funkcji dla szybkiego rozwoju aplikacji webowych. Jego wsparcie dla wzorców projektowych, jak MVC, oraz wbudowane mechanizmy autentykacji i ochrony sesji, sprawiają, że jest idealny do budowy bezpiecznego API backendowego.
React.js	Biblioteka JavaScript od Facebooka, znana z efektywnego renderowania UI i obsługi stanu aplikacji. Wybrana ze względu na modułowość, szeroką społeczność oraz łatwość integracji z różnymi backendami. Idealna do tworzenia dynamicznych, responsywnych interfejsów użytkownika.
PostgreSQL	Zaawansowany system zarządzania relacyjnymi bazami danych, który oferuje wsparcie dla dużych obciążeń i skomplikowanych zapytań. Wybrany dla stabilności, wydajności oraz doskonałego zarządzania transakcjami, co jest kluczowe w aplikacjach wymagających niezawodnego przechowywania danych.
Docker	Platforma do konteneryzacji aplikacji, która umożliwia łatwe pakowanie, dystrybucję i zarządzanie aplikacjami w izolowanych środowiskach. Docker jest wykorzystywany do usprawnienia procesów wdrażania oraz zapewnienia spójności środowisk od rozwoju po produkcję.

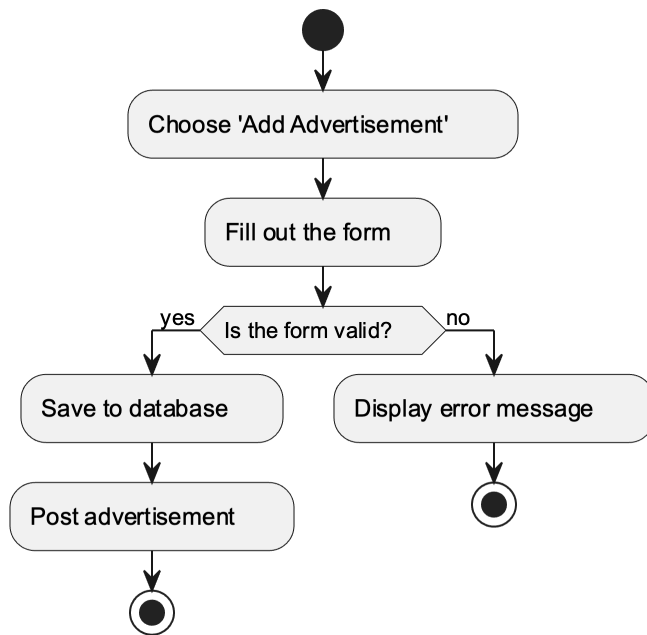
8.3 Diagramy UML

○ Diagram klas



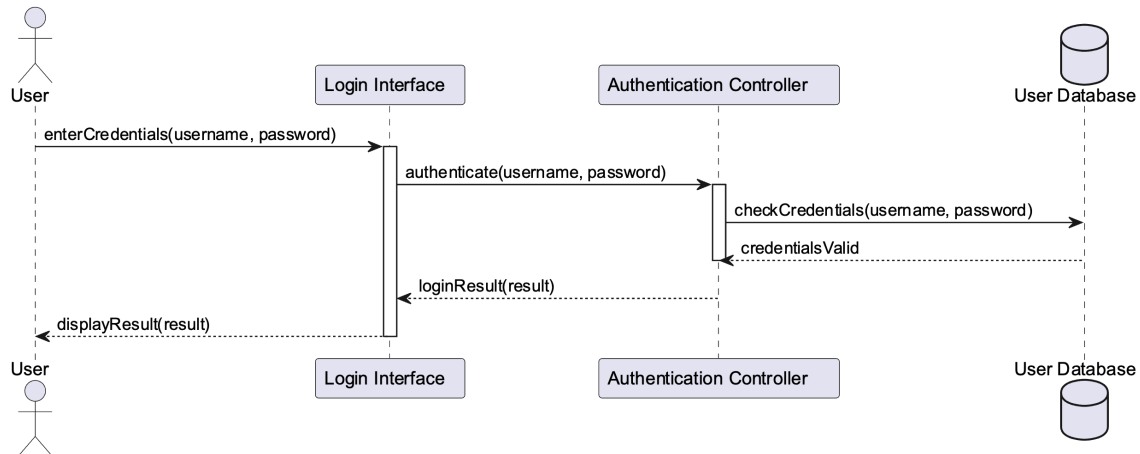
○ Diagramy czynności

Diagram dodawania:



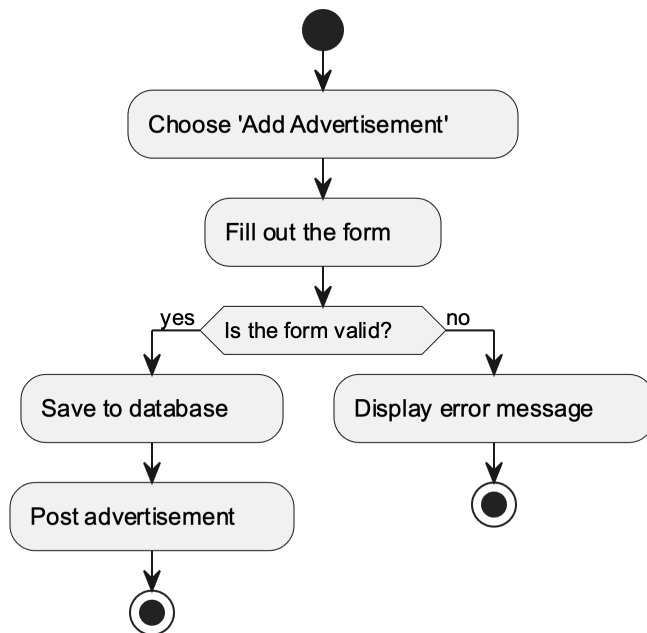
○ Diagramy sekwencji

Logowanie:

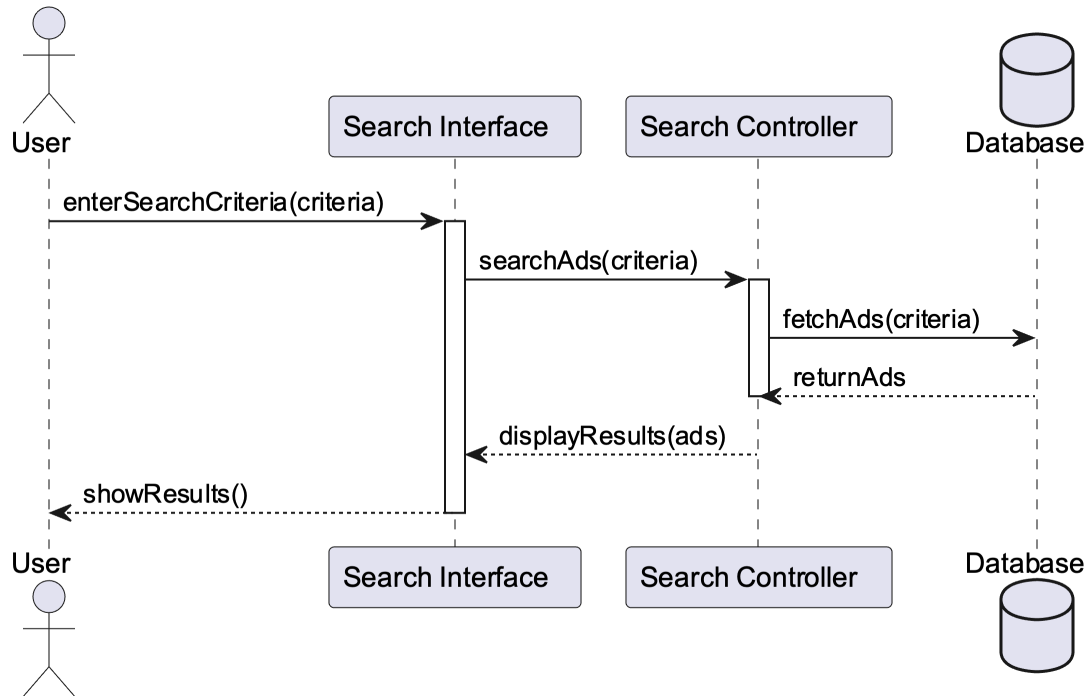


CRUD (Create, Read, Update, Delete) ogłoszeń:

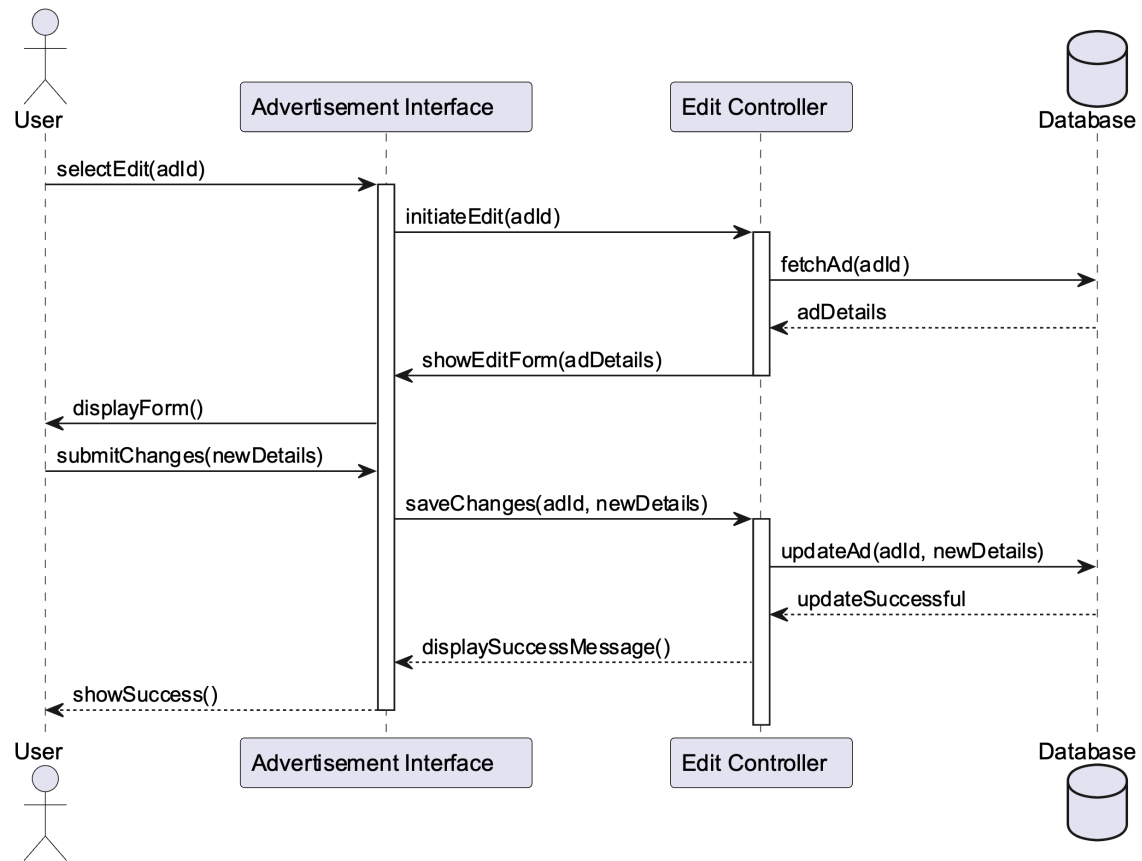
1. Dodawanie:



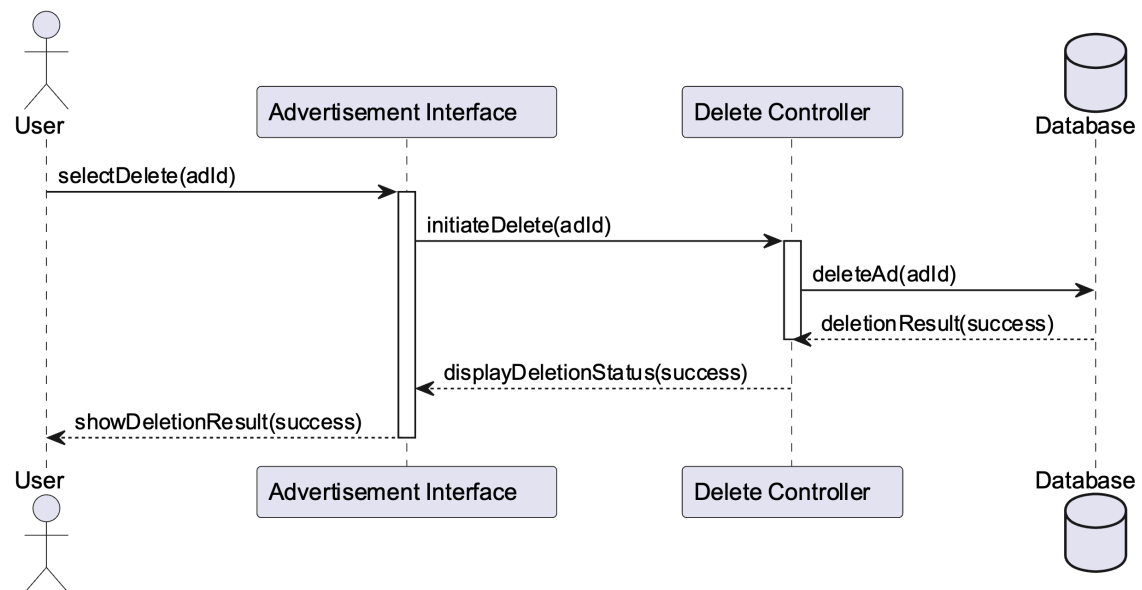
2. Odczyt (wyszukiwanie):



3. Edycja:

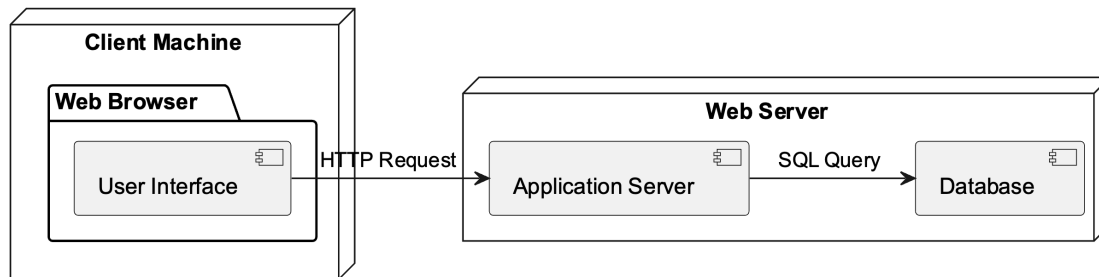


4. Usuwanie:



○ Inne diagramy

Diagram rozmieszczania:



7.4 Charakterystyka zastosowanych wzorców projektowych

W projekcie "Lokalizator" wykorzystano kilka kluczowych wzorców projektowych, które pomagają w organizacji kodu, ułatwiają jego rozwój oraz konserwację, a także zwiększają jego czytelność i skalowalność. Laravel jako framework PHP wspiera i promuje stosowanie wzorców projektowych, co znacznie ułatwia implementację skomplikowanych funkcjonalności. Poniżej przedstawiamy charakterystykę głównych wzorców projektowych zastosowanych w tym projekcie.

1. Wzorzec MVC (Model-View-Controller)

Opis: MVC jest podstawowym wzorcem projektowym używanym w Laravelu. Podzielony jest na trzy główne komponenty:

- **Model:** Reprezentuje dane aplikacji oraz logikę biznesową. W Laravelu, model jest odpowiedzialny za interakcję z bazą danych, zazwyczaj za pomocą Eloquent ORM.
- **View:** Odpowiada za prezentację danych użytkownikowi. W Laravelu, widoki są tworzone za pomocą Blade, systemu szablonów, który ułatwia generowanie HTML.
- **Controller:** Pośredniczy między modelem a widokiem, obsługując logikę aplikacji oraz żądania użytkowników. Kontroler przetwarza dane z modelu i przygotowuje je do wyświetlenia w widoku.

Zalety:

- Ułatwia organizację kodu poprzez separację logiki biznesowej, logiki prezentacji i logiki aplikacji.
- Poprawia czytelność i konserwowalność kodu.
- Umożliwia równoczesną pracę wielu programistów nad różnymi warstwami aplikacji.

2. Wzorzec Singleton

Opis: Wzorzec Singleton zapewnia, że dana klasa ma tylko jedną instancję i zapewnia globalny punkt dostępu do tej instancji. W Laravelu singleton jest często używany w kontekście kontenerów usług.

Przykład zastosowania:

- **Service Container:** Laravel wykorzystuje wzorzec singleton do zarządzania instancjami klas w kontenerze usług. Dzięki temu można łatwo kontrolować cykl życia obiektów i zależności między nimi.

Zalety:

- Redukuje zużycie pamięci, zapewniając pojedynczą instancję klasy.
- Ułatwia zarządzanie zależnościami w aplikacji.

3. Wzorzec Dependency Injection

Opis: Wzorzec Dependency Injection polega na przekazywaniu zależności do klasy zamiast tworzenia ich wewnątrz klasy. Laravel intensywnie korzysta z tego wzorca poprzez swój kontener usług.

Przykład zastosowania:

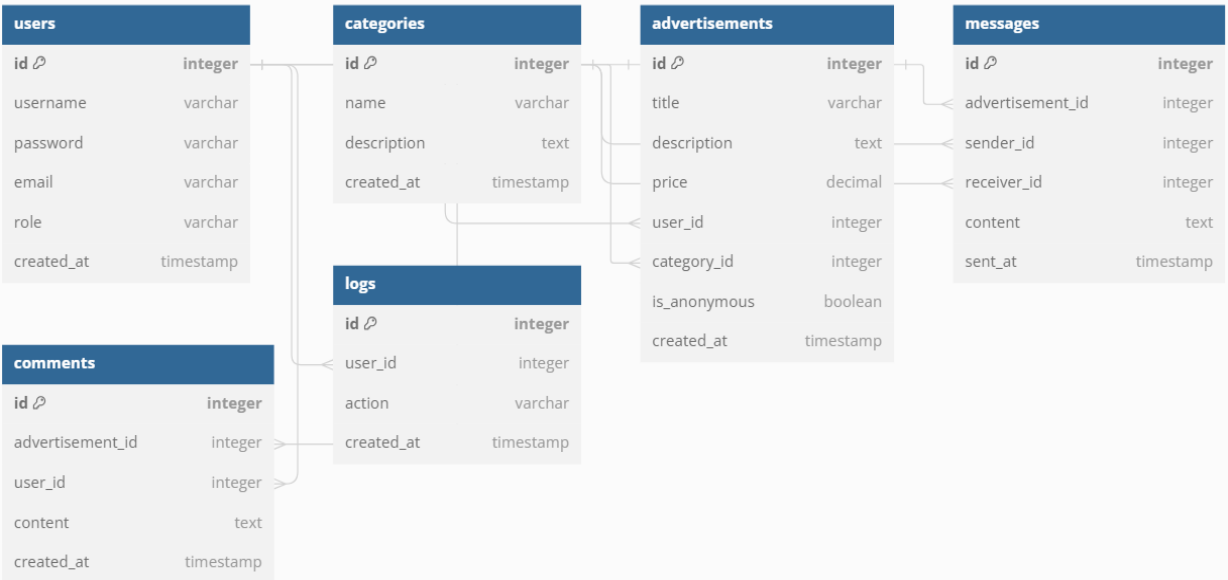
- **Kontener usług:** Laravel umożliwia rejestrowanie zależności w kontenerze usług, co pozwala na ich automatyczne wstrzykiwanie do klas, które ich potrzebują.

Zalety:

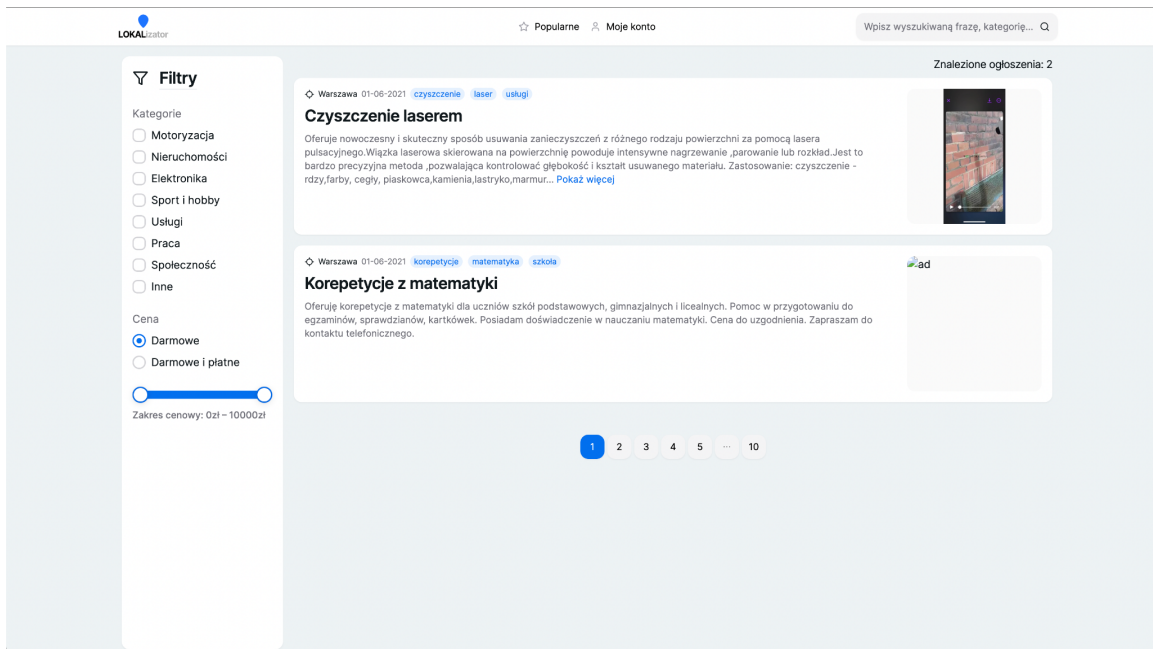
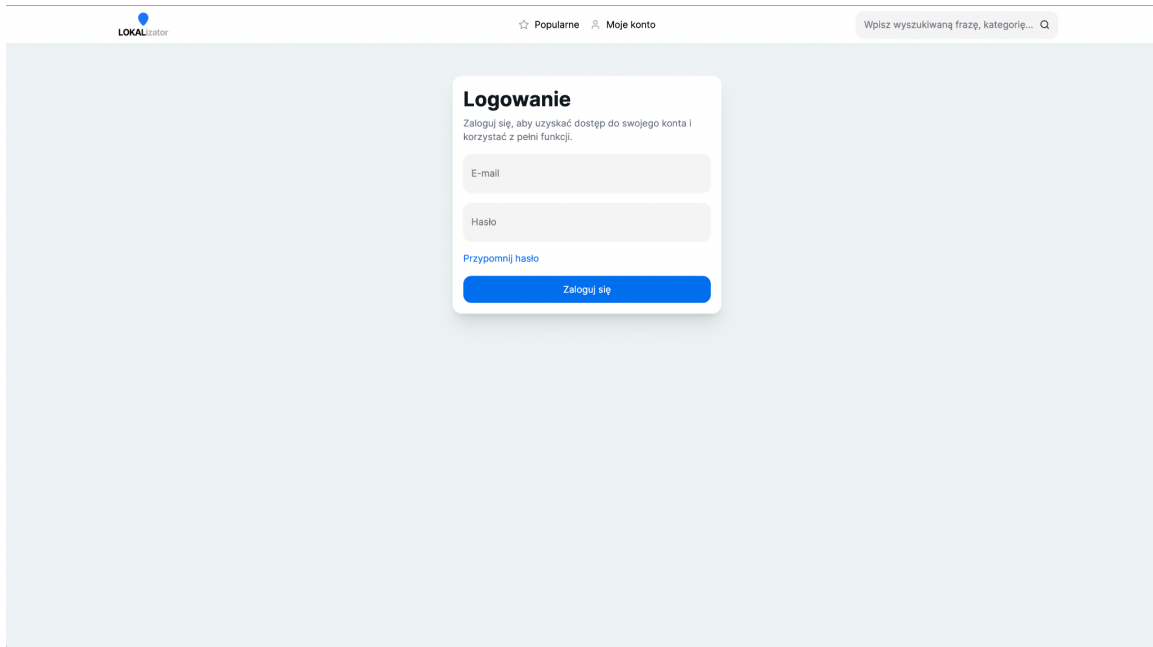
- Ułatwia zarządzanie zależnościami i ich testowanie.
- Zwiększa modularność i elastyczność kodu.

Te wzorce projektowe stanowią fundament dla architektury systemu "Lokalizator", zapewniając, że jest on dobrze zorganizowany, łatwy do utrzymania i skalowalny. Wykorzystanie tych wzorców pozwala na tworzenie efektywnego, bezpiecznego i wydajnego oprogramowania, które spełnia wysokie standardy jakości.

7.5 *Projekt* *bazy* *danych*



7.6 Projekt interfejsu użytkownika



Lista głównych elementów interfejsu

Ekrany:

1. Ekran logowania: Formularz logowania użytkownika z polami na adres e-mail i hasło oraz przyciskiem "Zaloguj się".
2. Ekran rejestracji: Formularz rejestracji nowego użytkownika z polami na adres e-mail, hasło, potwierdzenie hasła oraz przyciskiem "Zarejestruj się".
3. Ekran główny: Widok główny aplikacji z listą dostępnych ogłoszeń.
4. Ekran dodawania ogłoszenia: Formularz do tworzenia nowego ogłoszenia, zawierający pola na tytuł, opis, cenę, kategorię oraz możliwość dodania zdjęć.
5. Ekran przeglądania ogłoszenia: Widok szczegółów wybranego ogłoszenia z możliwością kontaktu z ogłoszeniodawcą.
6. Ekran edycji ogłoszenia: Formularz do edycji istniejącego ogłoszenia.
7. Ekran ustawień konta: Widok umożliwiający użytkownikowi edycję danych konta, takich jak adres e-mail, hasło itp.

Strony:

1. Strona wyszukiwania: Strona z funkcją wyszukiwania ogłoszeń według słów kluczowych i filtrów (lokalizacja, kategoria, cena).
2. Strona ulubionych ogłoszeń: Strona z listą ogłoszeń dodanych do ulubionych przez użytkownika.
3. Strona kontaktu: Strona z formularzem kontaktowym umożliwiającym bezpośredni kontakt z obsługą aplikacji.
4. Strona pomocy: Strona z często zadawanymi pytaniami (FAQ) i instrukcjami użytkownika aplikacji.

Aktywności:

1. Aktywność logowania: Obsługuje ekran logowania użytkownika.
2. Aktywność rejestracji: Obsługuje proces rejestracji nowego użytkownika.
3. Aktywność główna: Obsługuje główny widok aplikacji z listą ogłoszeń.
4. Aktywność dodawania ogłoszenia: Obsługuje tworzenie nowego ogłoszenia.
5. Aktywność przeglądania ogłoszenia: Obsługuje wyświetlanie szczegółów ogłoszenia.
6. Aktywność edycji ogłoszenia: Obsługuje edycję istniejącego ogłoszenia.
7. Aktywność ustawień konta: Obsługuje zarządzanie danymi konta użytkownika.
8. Aktywność wyszukiwania: Obsługuje funkcję wyszukiwania ogłoszeń.

8. Podsumowanie

Szczegółowe nakłady projektowe członków zespołu

	Przemysław Niemiec	Patryk Surmacz	Czas %
Dokumentacja punkt 5	80%	20%	9
Dokumentacja punkt 6	100%	0%	4
Dokumentacja punkt 8	100%	0%	6
Dokumentacja punkt 9	10%	90%	8
Dokumentacja punkt 11	0%	100%	3
Backend	100%	0%	35
Frontend	0%	100%	35
Podsumowanie	53%	47%	