

# Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

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

*Należy pozostawić wszelkie nagłówki tego dokumentu, a umieszczać treść w odpowiednich miejscach zamiast obecnych objaśnień.*

*Stronę tytułową można sformatować w dowolny sposób, ale należy pozostawić zawartość informacyjną w układzie pokazanym powyżej.*

*Praca powinna zostać złożona wyłącznie w formacie pdf. Przed wygenerowaniem ostatecznej wersji należy zaktualizować spis treści – wyświetlane dwa poziomy.*

*Niniejszą informację należy również usunąć z wersji końcowej.*

# 1 Spis treści

2	Odnośniki do innych źródeł.....	4
3	Słownik pojęć.....	5
4	Wprowadzenie.....	6
4.1	Cel dokumentacji.....	6
4.2	Przeznaczenie dokumentacji.....	6
4.3	Opis organizacji lub analiza rynku.....	6
4.4	Analiza SWOT organizacji.....	6
5	Specyfikacja wymagań.....	7
5.1	Charakterystyka ogólna.....	7
5.2	Wymagania funkcjonalne.....	8
5.3	Wymagania niefunkcjonalne.....	13
6	Zarządzanie projektem.....	15
6.1	Zasoby ludzkie.....	15
6.2	Harmonogram prac.....	15
6.3	Etapy/kamienie milowe projektu.....	15
7	Zarządzanie ryzykiem.....	17
7.1	Lista czynników ryzyka.....	17
7.2	Ocena ryzyka.....	17
7.3	Plan reakcji na ryzyko.....	17
8	Zarządzanie jakością.....	18
8.1	Scenariusze i przypadki testowe.....	18
9	Projekt techniczny.....	23
9.1	Opis architektury systemu.....	23
9.2	Technologie implementacji systemu.....	23
9.3	Diagramy UML.....	23
9.4	Charakterystyka zastosowanych wzorców projektowych.....	28
9.5	Projekt bazy danych.....	28
9.6	Projekt interfejsu użytkownika.....	29
9.7	Procedura wdrożenia.....	29
10	Dokumentacja dla użytkownika.....	31
11	Podsumowanie.....	32
11.1	Szczegółowe nakłady projektowe członków zespołu.....	32
12	Inne informacje.....	33



## **Odnosniki do innych źródeł**

tj. do wykorzystywanych narzędzi / projektów w tych narzędziach

- Zarządzania projektem – Jira, Trello, itp.
- Wersjonowanie kodu – sugerowany Git (hosting np. na Bitbucket lub Github), ew. SVN
- System obsługi defektów – np. Bitbucket, Github, Bugzilla.

GITHUB:

[https://github.com/vinze-ps/collage\\_software\\_engineering\\_LOKALizator](https://github.com/vinze-ps/collage_software_engineering_LOKALizator)

Trello:

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

# **Wprowadzenie**

## ***1.1 Cel dokumentacji***

po co ją robimy i co zawiera (poziom szczegółowości)

## ***Przeznaczenie dokumentacji***

dla kogo ona jest

## ***Opis organizacji lub analiza rynku***

Jedna z dwóch opcji:

1. Jeśli dla konkretnej organizacji: Czym jest organizacja, dla której realizowany będzie system; jak działa (lub będzie działała), kiedy system będzie wdrożony – tutaj nie odwołujemy się do samego systemu, tylko opisujemy samo działanie organizacji i role. W szczególności – jak wyglądają główne procesy biznesowe.
2. Jeśli na masowy rynek: Pobieźna analiza rynku. Dla kogo będzie przydatny taki system. Ile jest organizacji, które będą mogły z niego skorzystać, użytkowników w poszczególnych organizacjach. Czy te organizacje stanowią jednorodną grupę czy są różne rodzaje. Co one mają ze sobą wspólnego. Jak ta liczba będzie się zmieniała w najbliższej przyszłości.

## ***Analiza SWOT organizacji***

- jeśli system dla konkretnej organizacji:
  - o wystarczy sama tabela 2x2 (silne-słabe-szanse-zagrożenia)
- jeśli system na masowy rynek:
  - o szanse i zagrożeniaC

## **2 Specyfikacja wymagań**

### **2.1 Charakterystyka ogólna**

#### **2.1.1 Definicja produktu**

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

#### **2.1.2 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.

#### **2.1.3 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.

#### **2.1.4 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

#### **2.1.5 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.

### 2.1.6 Ograniczenia projektowe i wdrożeniowe

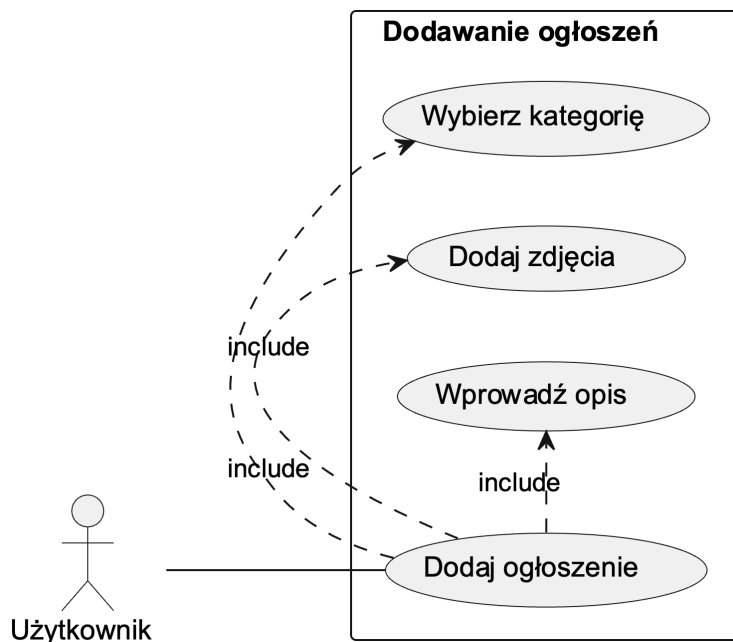
Projekt musi być zgodny z RODO w zakresie ochrony danych osobowych, co wymaga implementacji odpowiednich mechanizmów bezpieczeństwa i zasad prywatności. System musi być również dostosowany do obsługi na różnych platformach (desktop, mobile), co wymaga responsywnego projektowania interfejsów. Wymagane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

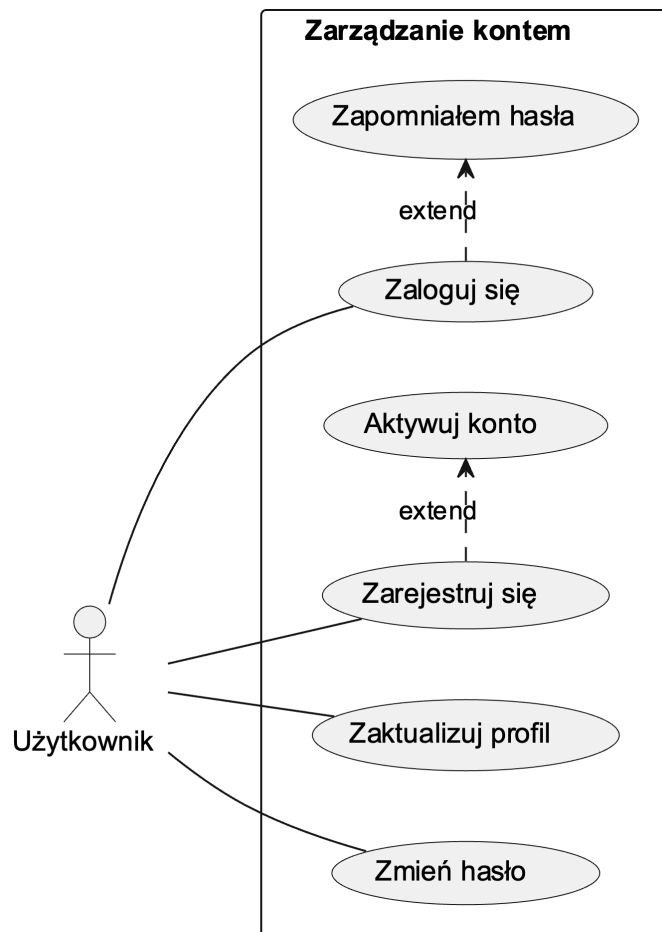
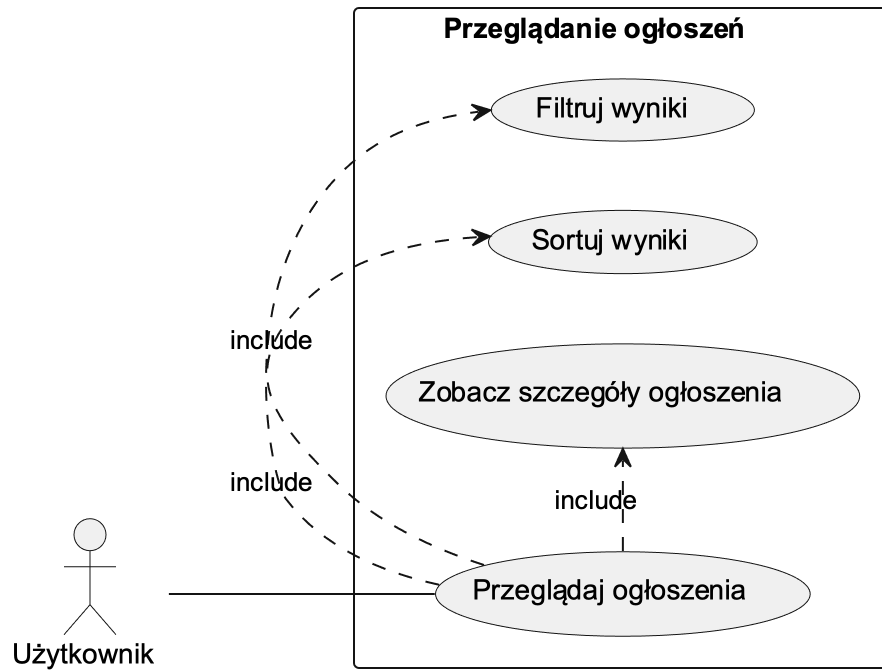
## 2.2 Wymagania funkcjonalne

### 2.2.1 Lista wymagań

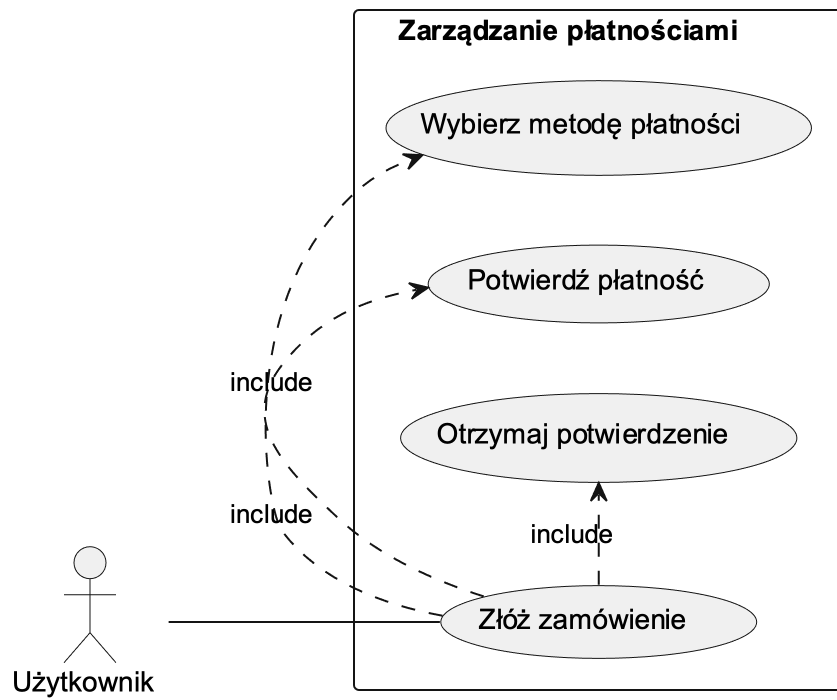
1. Użytkownik może tworzyć konto i zarządzać nim.
2. Użytkownik może dodawać, edytować i usuwać ogłoszenia.
3. Użytkownik może przeglądać ogłoszenia według kategorii.
4. Użytkownik może wyszukiwać ogłoszenia używając filtrów (np. lokalizacja, rodzaj ogłoszenia, cena).
5. Użytkownik może dodawać ogłoszenia do ulubionych.
6. System oferuje opcje kontaktu między kupującymi a sprzedającymi.
7. System umożliwia zarządzanie płatnościami w przypadku usług premium.

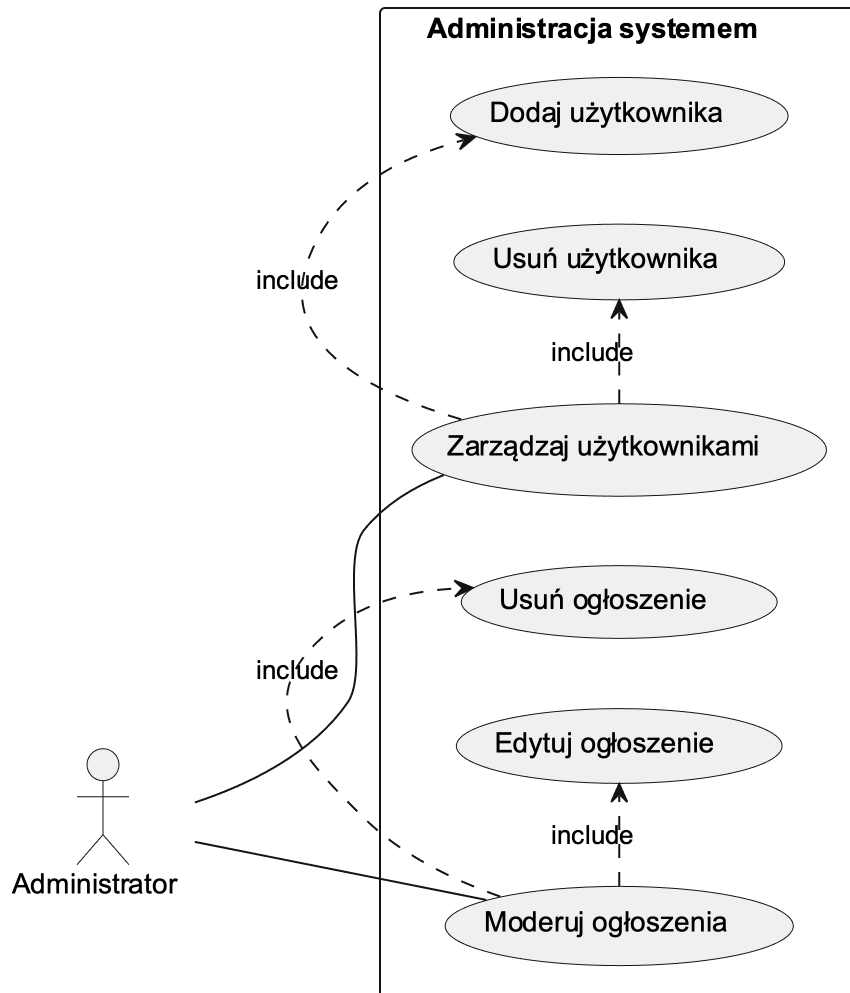
### 2.2.2 Diagramy przypadków użycia











## 2.2.3 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 niefunkcjonalne:** System powinien obsłużyć żądanie dodania pojedynczego ogłoszenia 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 niefunkcjonalne:** 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 niefunkcjonalne:** 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 niefunkcjonalne:** 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)

## **2.3 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.

## **3 Zarządzanie projektem**

### **3.1 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 ostatecznego produktu. (1 osoba)
- Quality Assurance Specialist: odpowiedzialny za testowanie całego systemu, zapewnienie jakości i spełnienie wymagań funkcjonalnych i нефункциональных. (1 osoba)
- UX/UI Designer: odpowiedzialny za projektowanie estetyczne i funkcjonalne interfejsu użytkownika, zapewniając przyjazne doświadczenie dla użytkownika końcowego. (1 osoba)

### **3.2 Harmonogram prac**

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

### **3.3 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.



# 4 Zarządzanie jakością

## 4.1 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:**
  - o Użytkownik musi być zalogowany.
  - o 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:**
  - o W systemie istnieją ogłoszenia spełniające kryteria wyszukiwania.
  - o 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:**
  - o Użytkownik jest zalogowany.
  - o 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:**
  - o Użytkownik jest zalogowany.
  - o 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.

### Test Case 5: Dodawanie Anonimowego Ogłoszenia

- **ID:** TC-05
- **Nazwa scenariusza:** Dodawanie Anonimowego Ogłoszenia
- **Kategoria:** Test funkcjonalny
- **Opis:** Sprawdzenie, czy użytkownik może dodać ogłoszenie bez logowania.
- **Tester:** Monika Zielińska
- **Termin:** 2024-06-05
- **Narzędzia wspomagające:** Postman, Selenium
- **Założenia, środowisko, warunki wstępne, dane wejściowe:**
  - o Przeglądarka internetowa z dostępem do serwera testowego.

### Przebieg działań

Lp.	Działania testera	Oczekiwane działania systemu
1	Użytkownik wybiera opcję "Dodaj Anonimowe Ogłoszenie".	System wyświetla formularz dodawania ogłoszenia.
2	Użytkownik wypełnia formularz (tytuł, opis, zdjęcia).	System przyjmuje dane i weryfikuje poprawność pól.
3	Użytkownik klika przycisk "Zapisz".	System zapisuje ogłoszenie jako anonimowe i wyświetla potwierdzenie.

### Zestaw danych testowych

Lp.	Dane wejściowe	Oczekiwane dane wyjściowe
1	Tytuł: "Zgubiono portfel", Opis: "Czarny skórzany portfel", Zdjęcia: "portfel.jpg"	Komunikat: "Ogłoszenie zostało dodane pomyślnie"

**Warunek zaliczenia testu:** Anonimowe ogłoszenie zostaje dodane i jest widoczne w systemie.

## 5 Projekt techniczny

### 5.1 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.

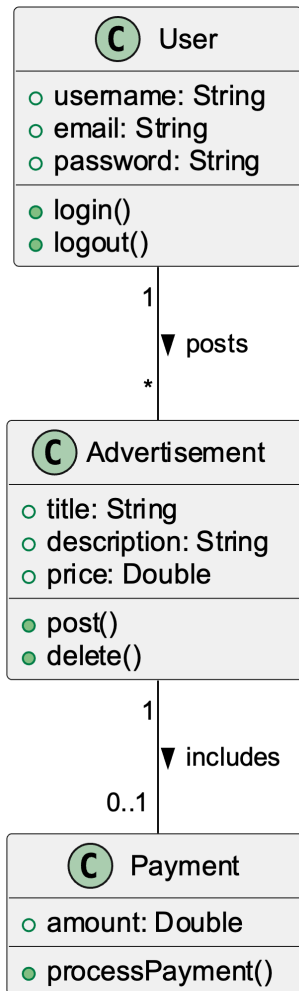
### 5.2 Technologie implementacji systemu

Technologia	Uzasadnienie wyboru
<b>Laravel 11</b>	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.
<b>React.js</b>	Biblioteka JavaScript od Facebooka, znana z efektywnego renderowania UI i obsługi stanu aplikacji. Wybrana ze względu na modularność, szeroką społeczność oraz łatwość integracji z różnymi backendami. Idealna do tworzenia dynamicznych, responsywnych interfejsów użytkownika.
<b>PostgreSQL</b>	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.
<b>Docker</b>	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.

### 5.3 Diagramy UML

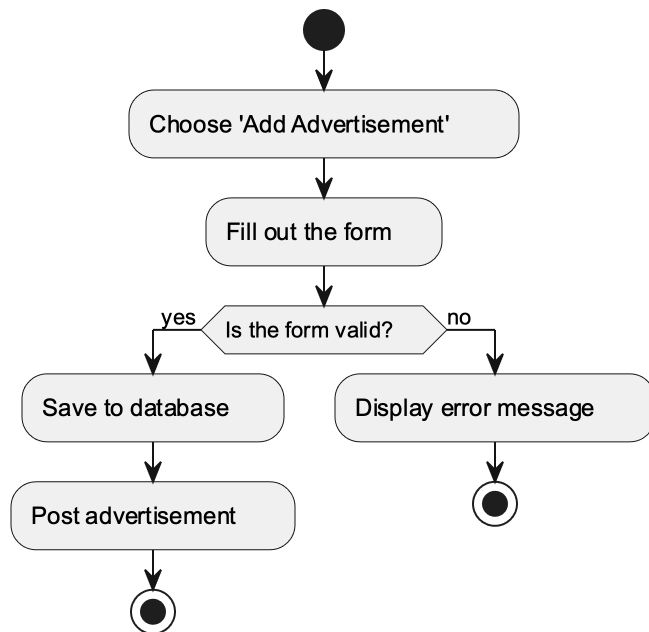
każdy diagram ma mieć tytuł oraz ma być na osobnej stronie  
diagramy przypadków użycia umieszczone w punkcie 4.2.2, a nie tutaj.

### 5.3.1 Diagram(-y) klas



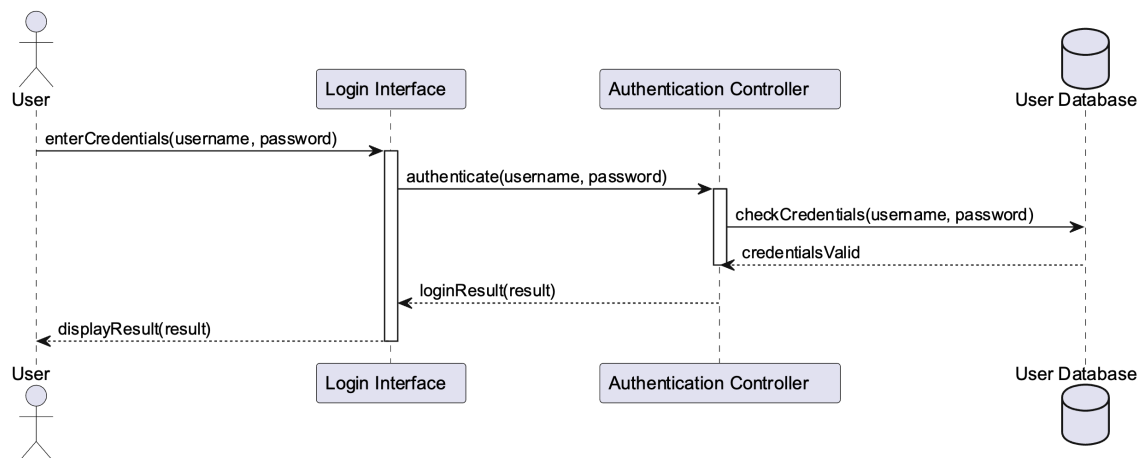
### 5.3.2 Diagram(-y) czynności

Diagram dodawania:



### 5.3.3 Diagramy sekwencji

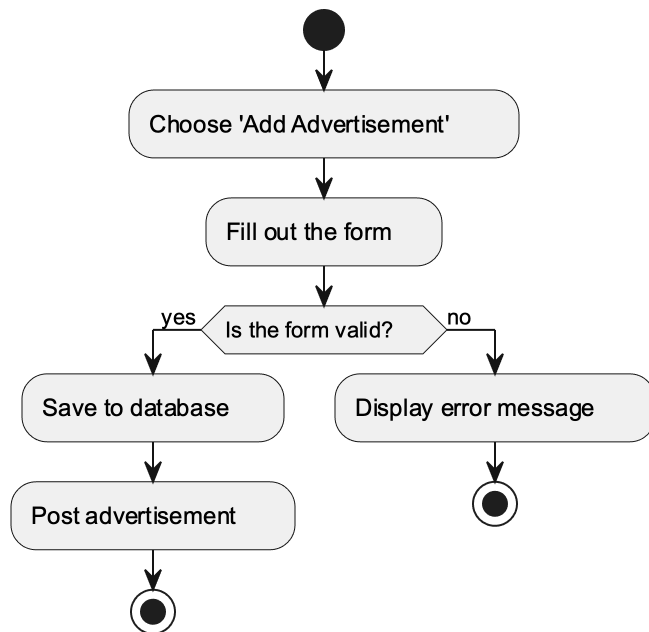
Logowanie:



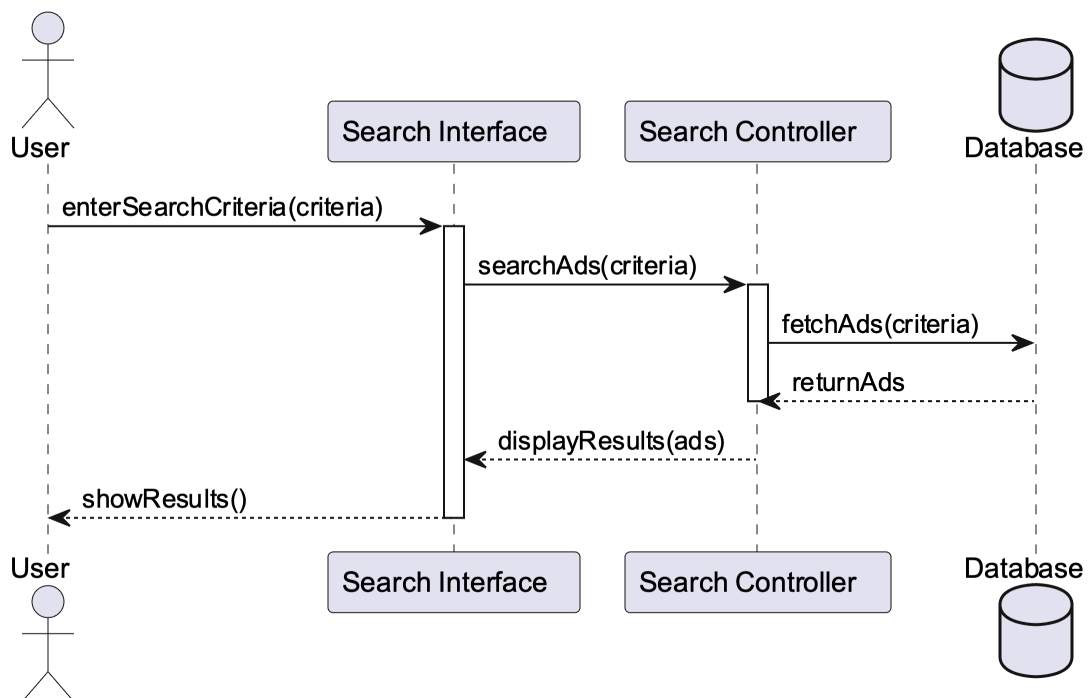
CRUD (Create, Read, Update, Delete) ogłoszeń:

1. Dodawanie:

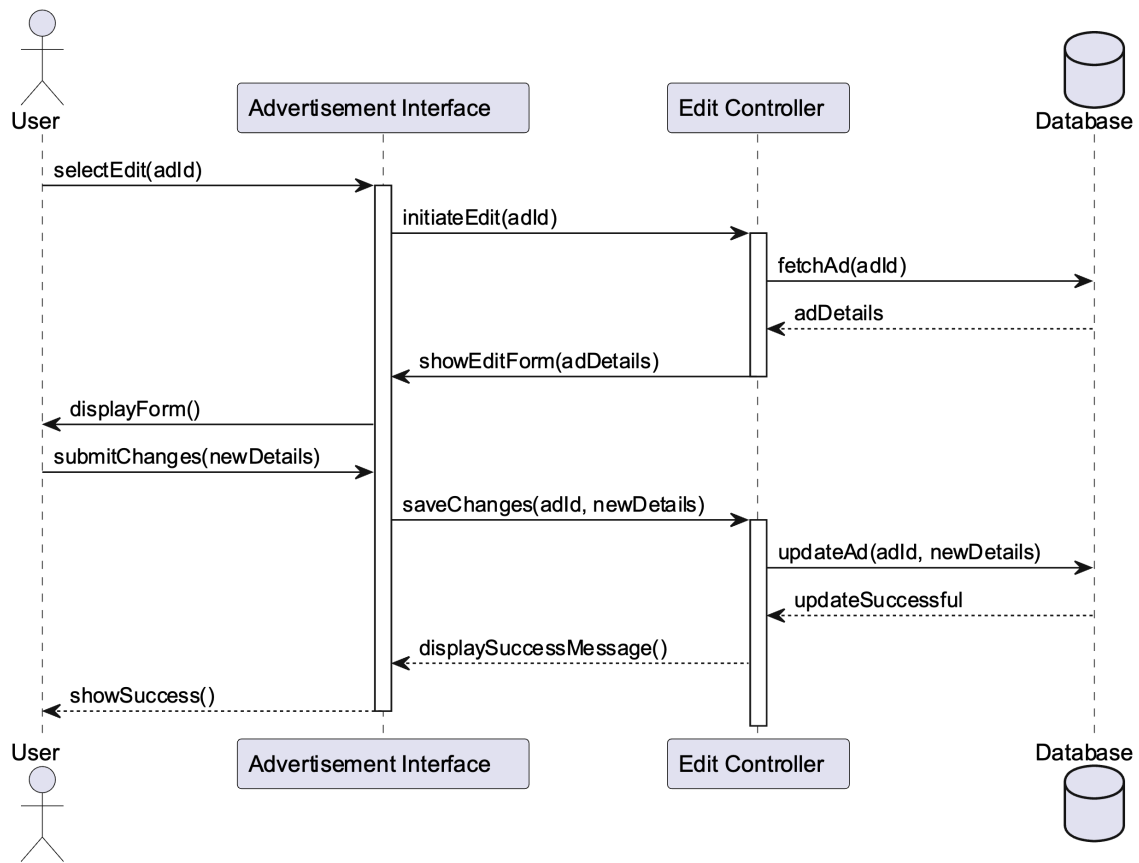




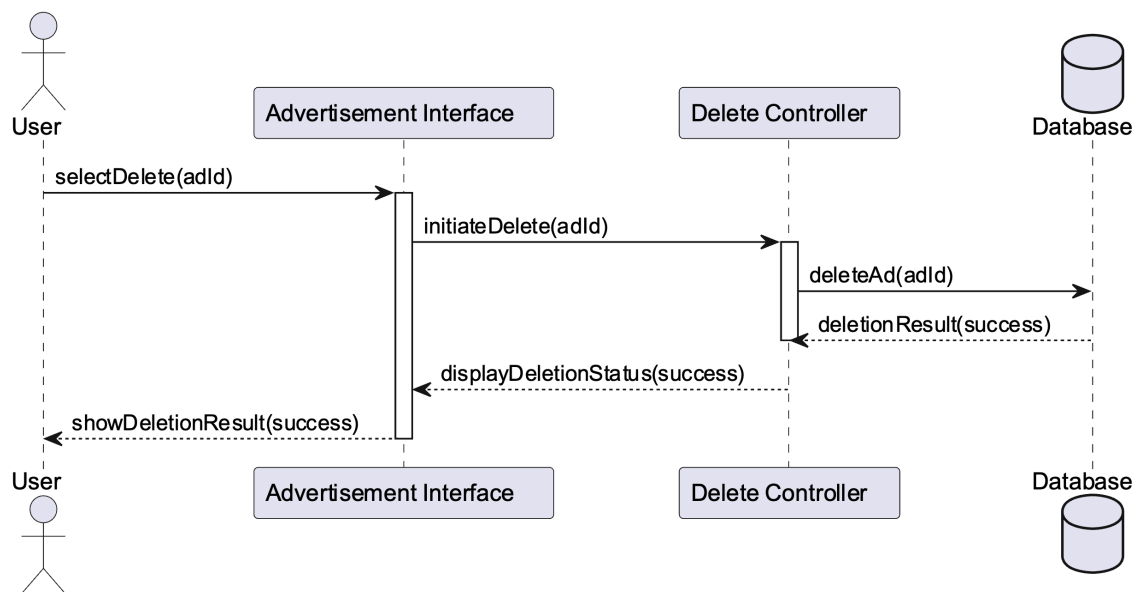
2. Odczyt (wyszukiwanie):



### 3. Edycja:

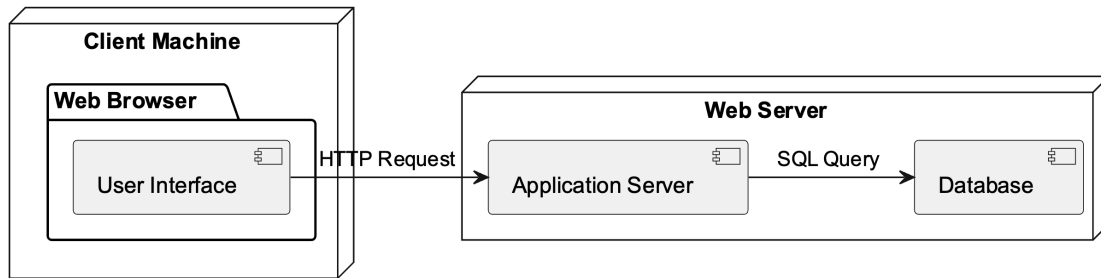


### 4. Usuwanie:



### 5.3.4 Inne diagramy

Diagram rozmieszczania:

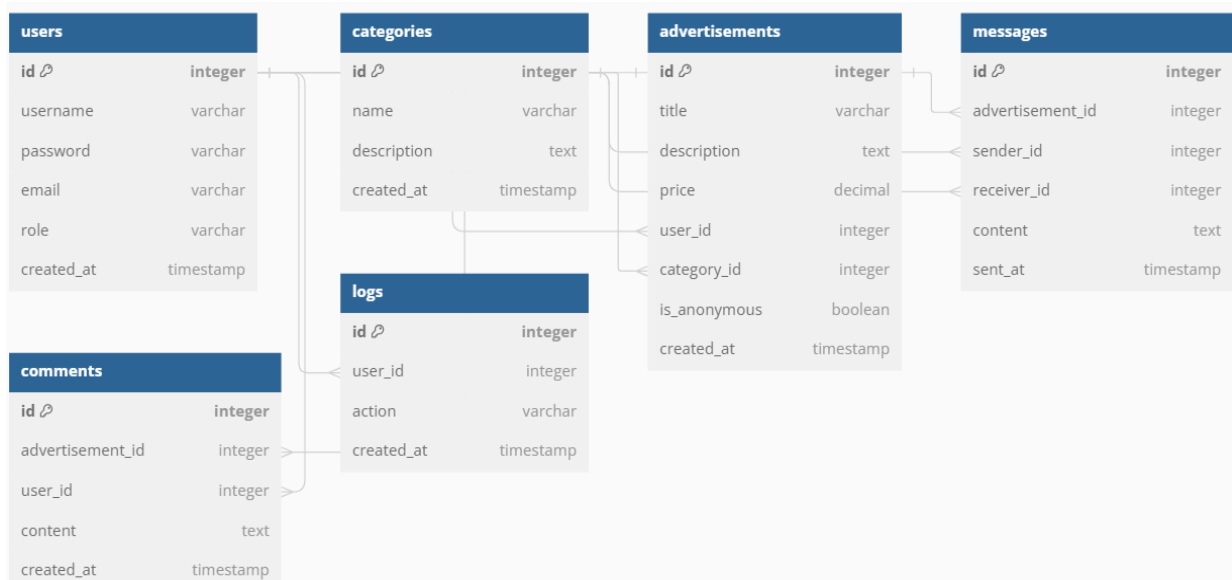


### 5.4 Charakterystyka zastosowanych wzorców projektowych

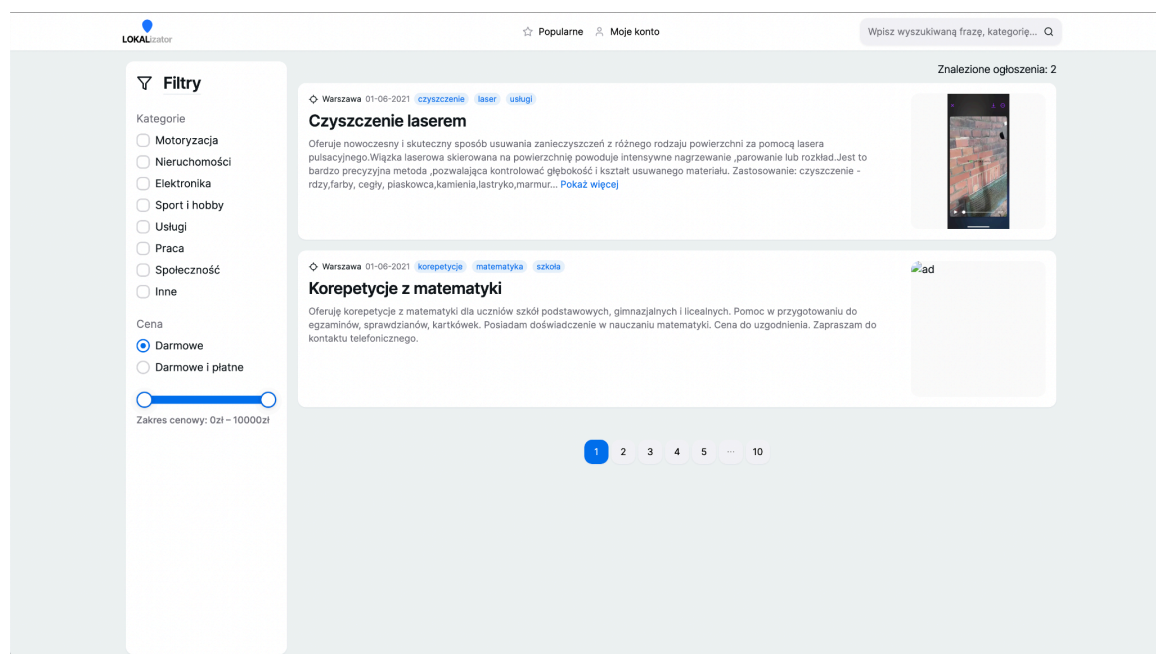
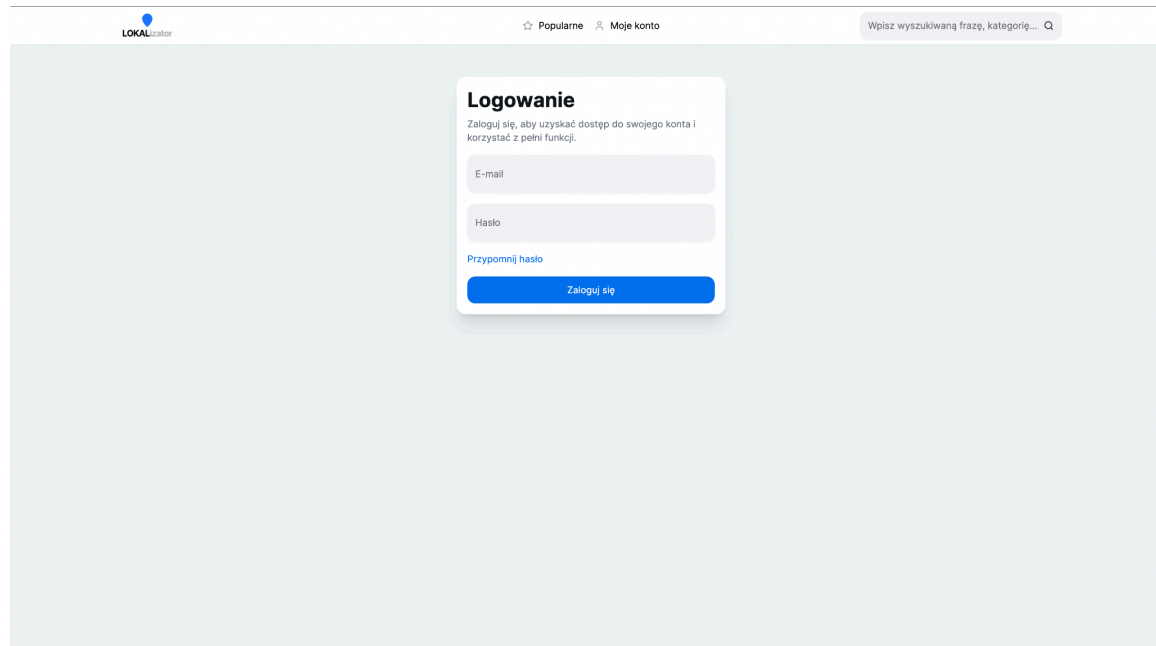
informacja opisowa wspomagana diagramami (odsylaczami do diagramów UML); jeśli wykorzystano wzorce projektowe, to należy wykazać dwa z nich  
uwaga – wzorce projektowe nie są omawiane na wykładach!

### 5.5 Projekt bazy danych

#### 5.5.1 Schemat



### 5.6 Projekt interfejsu użytkownika



## 5.6.1 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.

## **5.6.2 Przejścia między głównymi elementami**

Storyboard:

1. Przejście z ekranu logowania do ekranu głównego po pomyślnym zalogowaniu.
2. Przejście z ekranu rejestracji do ekranu logowania po pomyślnej rejestracji.
3. Przejście z ekranu głównego do ekranu dodawania ogłoszenia po kliknięciu przycisku "Dodaj Ogłoszenie".
4. Przejście z ekranu głównego do ekranu przeglądania ogłoszenia po kliknięciu na wybrane ogłoszenie.
5. Przejście z ekranu przeglądania ogłoszenia do ekranu edycji ogłoszenia po kliknięciu przycisku "Edytuj".
6. Przejście z ekranu głównego do ekranu ustawień konta po kliknięciu przycisku "Ustawienia".



## 6 Podsumowanie

### 6.1 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
<b>Podsumowanie</b>	<b>53%</b>	<b>47%</b>	

## **7   Inne informacje**

przydatne informacje, które nie zostały ujęte we wcześniejszych punktach