

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ INFORMATYKI I TELEKOMUNIKACJI

Internetowe Bazy Danych

Serwis ogłoszeń mieszkaniowych

ANALIZA I ZDEFINIOWANIE WYMAGAŃ DLA PROJEKTU

PROWADZĄCY:

dr inż. Roman Ptak

AUTORZY:

Jarosław Kopaczewski, 252712

Mikołaj Szymczyk, 248881

Daniel Glazer, 252743

Wrocław 12.12.2022

Spis treści

Spis treści	2
Spis rysunków	4
Spis tabel	5
1 Wstęp	6
1.1 Cel i zakres projektu	6
1.2 Harmonogram	6
2 Analiza wymagań	7
2.1 Wymagania funkcjonalne	7
2.2 Wymagania нефункционалне	8
2.3 Diagram przypadków użycia	8
2.4 Analiza biznesowa	11
2.4.1 Analiza kosztów	11
2.5 Analiza SWOT	13
2.5.1 Opis czynników	13
2.5.2 Analiza powiązań SWOT	13
2.5.3 Analiza powiązań TOWS	14
2.5.4 Wybór strategii	15
3 Projekt systemu	16
3.1 Architektura systemu	16
3.2 Technologie	17
3.2.1 Techniczna wykonalność projektu	17
3.3 Model logiczny bazy danych	17
3.4 Model fizyczny bazy danych	19
4 Implementacja	20
4.1 Perspektywa klienta	20

4.1.1	Widoki aplikacji	20
4.2	Panel administratora	23
4.3	Perspektywa programisty	24
4.3.1	Punkty dostępu do serwera	24
4.3.2	Przykładowa implementacja punktu dostępu	24
4.3.3	Schemat połączenia punktów dostępu i bazy danych	25
4.3.4	Zabezpieczenie dostępu do endpointów	25
4.3.5	Generowanie ciasteczka dla zalogowanego użytkownika	26
4.3.6	Struktura projektu strony użytkownika	27
4.4	Perspektywa wdrożeniowca	29
5	Podsumowanie	30

Spis rysunków

2.1	Diagram przypadków użycia	8
2.2	Czy określona mocna strona pozwala wykorzystać daną szansę?	13
2.3	Czy określona mocna strona pozwala ograniczyć dane zagrożenie?	14
2.4	Czy określona słaba strona ogranicza możliwość wykorzystania danej szansy?	14
2.5	Czy określona słaba strona potęguje dane zagrożenie?	14
2.6	Czy określona szansa potęguje daną silną stronę?	14
2.7	Czy określone zagrożenie ogranicza daną silną stronę?	14
2.8	Czy określona szansa pozwala osłabić (przewyciężyć) daną słabą stronę?	15
2.9	Czy określone zagrożenie wzmacnia daną słabą stronę?	15
2.10	Wyniki zbiorcze analizy SWOT/TOWS	15
3.1	Architektura aplikacji trójwarstwowej	16
3.2	Model logiczny bazy danych	18
3.3	Diagram encji bazy danych	19
4.1	Główny widok aplikacji	21
4.2	Widoki aplikacji: a) Logowanie, b) Rejestracja	21
4.3	Szczegóły ogłoszenia	22
4.4	Ogłoszenia użytkownika	22
4.5	Dodanie ogłoszenia	23
4.6	Usuwanie użytkowników	23
4.7	Diagram kluczowych klas w dostępie do danych bazy danych	25
4.8	Struktura projektu strony użytkownika	28

Spis tabel

2.1	Koszty licencji	12
2.2	Koszty hostowania	12
2.3	Czynniki zewnętrzne analizy SWOT	13
2.4	Czynniki wewnętrzne analizy SWOT	13

Rozdział 1

Wstęp

1.1 Cel i zakres projektu

Zaprojektowanie i implementacja serwisu umożliwiającego publikację i przeglądanie ogłoszeń mieszkaniowych. Głównym założeniem projektowym jest dostarczenie intuicyjnej strony internetowej tworzącej połączenie pomiędzy kupującymi oraz sprzedającymi mieszkań. Aplikacja powinna składać się z części serwerowej oraz warstwy graficznej widocznej przez użytkownika.

1.2 Harmonogram

- 14.11.2022 - Analiza i zdefiniowanie wymagań dla projektu
- 28.11.2022 - Projekt, implementacja i wdrożenie systemu
- 12.12.2022 - Prezentacja projektu na forum publicznym wraz z oddaniem finalnej dokumentacji projektowej

Rozdział 2

Analiza wymagań

2.1 Wymagania funkcjonalne

Wymagania funkcjonalne dla niezalogowanego użytkownika

- przeglądanie ogłoszeń innych użytkowników na podstawie wybranych kryteriów takich, jak lokalizacja czy cena.
- zakładanie konta w systemie za pomocą formularza rejestracji.

Wymagania funkcjonalne dla zalogowanego użytkownika:

- posiadanie wszystkich funkcjonalności niezalogowanego użytkownika,
- tworzenie i edytowanie ogłoszeń,
- logowanie do systemu po wcześniejszym utworzeniu konta.

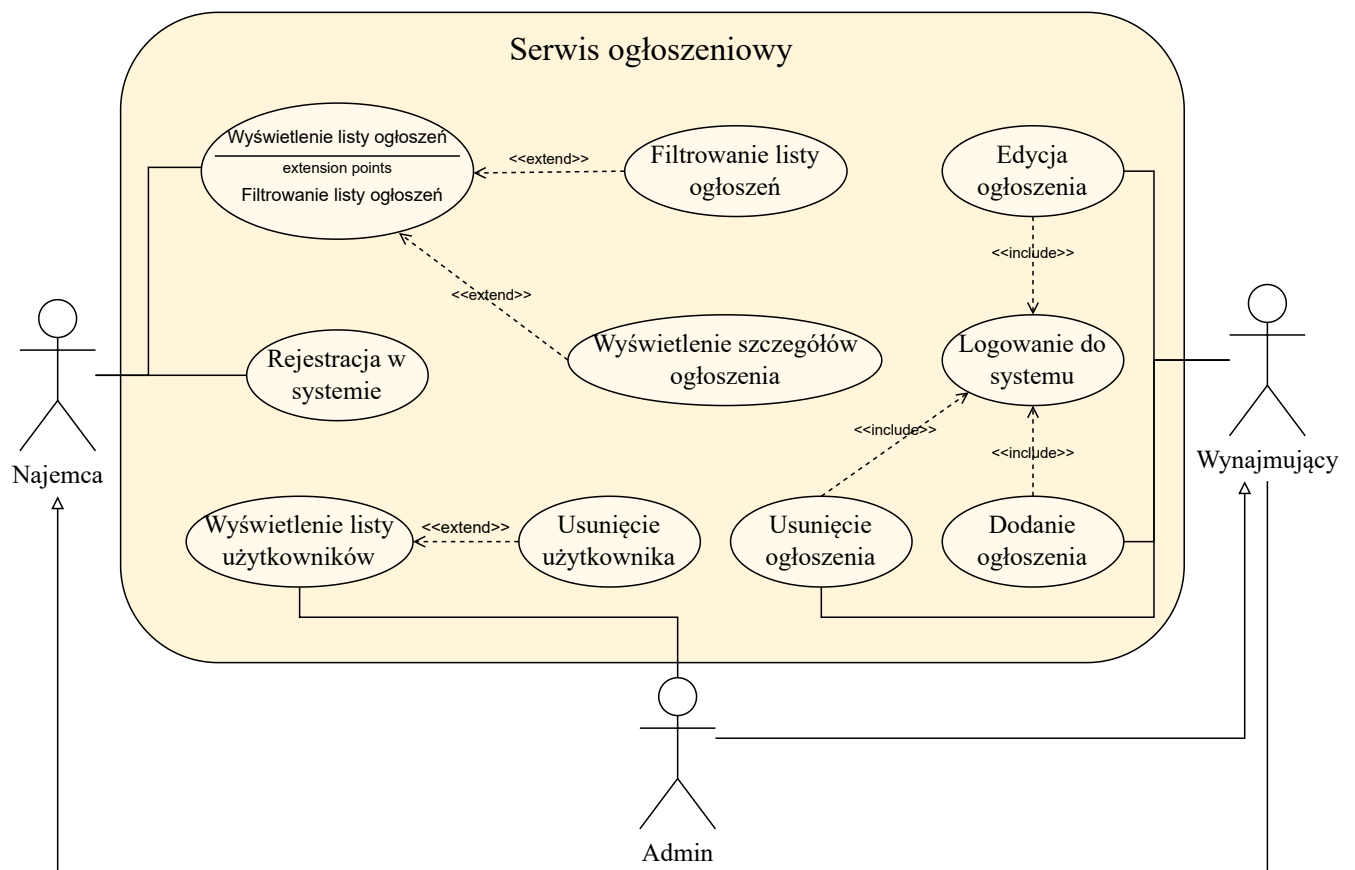
Wymagania funkcjonalne dla administratora:

- posiadanie wszystkich funkcjonalności użytkownika,
- usuwanie użytkowników,
- zarządzanie bazą danych,
- możliwość modyfikowania strony,
- śledzenie logów serwisu.

2.2 Wymagania niefunkcjonalne

- ogłoszenia mogą być wystawione tylko przez zalogowanego użytkownika,
- ogłoszenia mogą być przeglądane przez niezalogowanych użytkowników,
- statystyki ogłoszenia są dostępne tylko dla ogłoszeniodawcy,
- usunięcie użytkownika może nastąpić po wskazaniu punktu naruszającego regulamin serwisu.

2.3 Diagram przypadków użycia



Rysunek 2.1: Diagram przypadków użycia

PU-01: Rejestracja w systemie

Cel: Założenie konta umożliwiającego dodawanie ogłoszeń

Warunki początkowe: Kliknięcie przycisku „Zarejestruj się” w widoku logowania

Warunki końcowe: Dodanie konta użytkownika do bazy danych *Aktor:* Najemca

Przebieg:

1. Kliknięcie przycisku „Zarejestruj się”.

2. Uzupełnienie formularza poprawnymi danymi.
3. Kliknięcie przycisku „Zarejestruj się” w widoku rejestracji.

PU-02: Wyświetlenie listy ogłoszeń

Cel: Przeglądnięcie istniejących w serwisie ogłoszeń

Warunki początkowe: Otwarcie strony głównej serwisu

Warunki końcowe: Wyświetlenie listy wszystkich ogłoszeń *Aktor:* Najemca

Przebieg:

1. Kliknięcie przycisku „Wyszukaj” w widoku listy ogłoszeń.
2. Wyświetlenie listy ogłoszeń.
3. Może wywołać **PU-03 Filtrowanie listy ogłoszeń** oraz **PU-04 Wyświetlenie szczegółów ogłoszenia**.

PU-03: Filtrowanie listy ogłoszeń

Cel: Filtracja dostępnych ogłoszeń

Warunki początkowe: Może być wywołane przez **PU-02: Wyświetlenie listy ogłoszeń**

Warunki końcowe: Wyświetlenie listy ogłoszeń po filtracji *Aktor:* Najemca

Przebieg:

1. Uzupełnienie pól filtra takich jak np. Cena minimalna, cena maksymalna czy ilość pokoi.
2. Kliknięcie przycisku „Wyszukaj” w widoku listy ogłoszeń.

PU-04: Wyświetlenie szczegółów ogłoszenia

Cel: Wyświetlenie szczegółów wybranego ogłoszenia

Warunki początkowe: Może być wywołane przez **PU-02: Wyświetlenie listy ogłoszeń**

Warunki końcowe: Wyświetlenie ogłoszenia wybranego przez użytkownika

Aktor: Najemca

Przebieg:

1. Kliknięcie na wybrane ogłoszenie z listy.

PU-05: Logowanie do systemu

Cel: Zalogowanie się na stronie jako wynajmujący

Warunki początkowe: Wyświetlenie widoku logowania

Warunki końcowe: Zalogowanie do systemu

Aktor: Wynajmujący

Przebieg:

1. Uzupełnienie danych logowania i kliknięcie „Zaloguj się”.

PU-06: Edycja ogłoszenia

Cel: Zmiana zawartości ogłoszenia

Warunki początkowe: Wymaga wykonania **PU-05: Logowanie do systemu**

Warunki końcowe: Zmiana wartości pól ogłoszenia

Aktor: Wynajmujący

Przebieg:

1. Kliknięcie przycisku „Edytuj ogłoszenie”.
2. Wyświetlenie formularza z uzupełnionymi danymi.
3. Zmiana danych ogłoszenia.
4. Potwierdzenie zakończenia edycji lub anulowania edycji.

PU-07: Dodanie ogłoszenia

Cel: Dodanie nowego ogłoszenia

Warunki początkowe: Wymaga wykonania **PU-05: Logowanie do systemu** *Warunki końcowe:* Wprowadzenie ogłoszenia do systemu

Aktor: Wynajmujący

Przebieg:

1. Kliknięcie przycisku „Dodaj ogłoszenie” w widoku profilu użytkownika
2. Uzupełnienie formularza danymi
3. Dodanie zdjęć
4. Kliknięcie „Dodaj ogłoszenie”

PU-08: Usunięcie ogłoszenia

Cel: Usunięcie ogłoszenia przez użytkownika

Warunki początkowe: Wymaga wykonania **PU-05: Logowanie do systemu** *Warunki końcowe:* Usunięcie ogłoszenia z systemu

Aktor: Wynajmujący

Przebieg:

1. Wyświetlanie listy ogłoszeń użytkownika
2. Kliknięcie ikony „kosza” na ogłoszeniu do usunięcia.

PU-09: Usunięcie użytkownika

Cel: Usunięcie użytkownika naruszającego regulami serwisu

Warunki początkowe: Wymaga wykonania **PU-05: Logowanie do systemu** jako administrator

Warunki końcowe: Usunięcie użytkownika i jego ogłoszeń

Aktor: Admin

Przebieg:

1. Wybranie listowania użytkowników z menu bocznego.
2. Kliknięcie przycisku usuń obok pozycji wybranego użytkownika.

PU-10: Wyświetlenie listy użytkowników

Cel: Przeglądanie listy użytkowników dostępnych w systemie

Warunki początkowe: Wymaga wykonania **PU-05: Logowanie do systemu** jako administrator *Warunki końcowe:*

Usunięcie użytkownika i jego ogłoszeń *Aktor:* Admin

Przebieg:

1. Kliknięcie przycisku usuń na liście użytkowników.
2. Wyświetlenie listy użytkowników.

2.4 Analiza biznesowa

2.4.1 Analiza kosztów

Analiza zawiera szacowane koszty wynagrodzenia pracowników, licencji oprogramowania oraz hostingu gotowej aplikacji. W przypadku oprogramowania są to w większości rozwiązania open source. Dzięki temu jedyną licencją wymagającą wykupienia jest subskrypcja

IntelliJ/IDEA. Sumaryczny koszt oprogramowania oraz wynagrodzenia szacowany jest na 1020\$. Czas przewidziany na wykonanie projektu to 120h godzin roboczych.

Koszty realizacji projektu

Oprogramowanie	Koszt
IntelliJ IDEA	$3 \cdot 20\$$
Spring Boot	0\$
PostgreSQL	0\$
Java OpenJDK 11	0\$
Wynagrodzenie	$120h \cdot 8\$ = 960\$$

Tabela 2.1: Koszty licencji

Koszt hostingu

Koszty hostowania obejmują utrzymanie strony internetowej oraz bazy danych w serwisie MS Azure. Ceny są podane orientacyjnie. Mogą być większe w zależności od natężenia ruchu użytkowników oraz ilości przechowywanych w bazie danych.

Nazwa	Cena
Hostowanie strony	$0.10\$ \backslash h$
Hostowanie bazy	$0.20\$ \backslash h$
Rejestracja domeny	30\$

Tabela 2.2: Koszty hostowania

Sumaryczny koszt

Szacowany miesięczny koszt realizacji projektu wynosi 1020\$, który włącza wykupione licencje oraz wynagrodzenie pracowników. Wycena zakłada że pracownicy posiadają sprzęt umożliwiający realizację zadania. Natomiast orientacyjna cena utrzymania bazy danych oraz strony internetowej w serwisie MS Azure wynosi 216\$ miesięcznie. Wysokość kosztów związana jest z potrzebą przechowywania dużej ilości zdjęć ogłoszeń dodawanych przez użytkowników. Dodatkowo wymagana jest jednorazowa opłata mająca na celu rezerwację domeny oraz jej roczne opłacenie w wysokości 30\$.

2.5 Analiza SWOT

2.5.1 Opis czynników

Waga	Czynniki zewnętrzne
1,00	Szanse
0,40	Reklamy serwisu w innych aplikacjach.
0,15	Zwiększenie chęci na zainwestowanie pieniędzy, wynikającej z rosnącej inflacji.
0,15	Napływ imigrantów ze wschodniej granicy.
0,30	Szerokie grono odbiorców.
1,00	Zagrożenia
0,40	Istnieją już kilka podobnych produktów na rynku.
0,40	Tendencja spadkowa ilości dostępnych mieszkań.
0,20	Możliwy kryzys ekonomiczny.

Tabela 2.3: Czynniki zewnętrzne analizy SWOT

Waga	Czynniki wewnętrzne
1,00	Mocne strony
0,30	Specjalista od projektowania interfejsów graficznych.
0,25	Dobra znajomość wybranych technologii w zespole.
0,30	Zgrany zespół
0,15	Wykorzystanie nowych technologii zwiększa bezpieczeństwo serwisu.
1,00	Słabe strony
0,40	Brak specjalistów z dziedziny marketingu.
0,10	Brak doświadczenia w tworzeniu aplikacji trójwarstwowych.
0,10	Brak API do łączenia z wewnętrznymi aplikacjami biur nieruchomości.
0,40	Ograniczony czas na zrealizowanie projektu.

Tabela 2.4: Czynniki wewnętrzne analizy SWOT

2.5.2 Analiza powiązań SWOT

Mocne strony/ Szanse	[MS1]	[MS2]	[MS3]	[MS4]	Waga	Liczba interakcji	Iloczyn wag i interakcji	Ranga
[S1]	1	1	1	1	0,40	4	1,6	1
[S2]	0	1	1	0	0,15	2	0,3	3
[S3]	0	0	0	0	0,15	0	0	4
[S4]	1	1	1	0	0,30	3	0,9	2
Waga	0,30	0,25	0,30	0,15				
Liczba interakcji	2	3	3	1		18		
Iloczyn wag i interakcji	0,6	0,75	0,9	0,15			5,2	
Ranga	3	2	1	4				

Rysunek 2.2: Czy określona mocna strona pozwala wykorzystać daną szansę?

Mocne strony/ Zagrożenia	[MS1]	[MS2]	[MS3]	[MS4]	Waga	Liczba interakcji	Iloczyn wag i interakcji	Ranga
[Z1]	1	1	1	1	0,40	4	1,6	1
[Z2]	0	0	0	0	0,40	0	0	3
[Z3]	0	0	1	0	0,20	1	0,2	2
Waga	0,30	0,25	0,30	0,15				
Liczba interakcji	1	1	2	1		10		
Iloczyn wag i interakcji	0,3	0,25	0,6	0,15			3,1	
Ranga	2	3	1	4				

Rysunek 2.3: Czy określona mocna strona pozwala ograniczyć dane zagrożenie?

Słabe strony/Szanse	[SS1]	[SS2]	[SS3]	[SS4]	Waga	Liczba interakcji	Iloczyn wag i interakcji	Ranga
[S1]	1	1	0	0	0,40	2	0,8	1
[S2]	0	0	0	0	0,15	0	0	2
[S3]	0	0	0	0	0,15	0	0	2
[S4]	0	0	0	0	0,30	0	0	2
Waga	0,40	0,10	0,10	0,40				
Liczba interakcji	1	1	0	0		4		
Iloczyn wag i interakcji	0,4	0,1	0	0			1,3	
Ranga	1	2	3	3				

Rysunek 2.4: Czy określona słaba strona ogranicza możliwość wykorzystania danej szansy?

Słabe strony/Zagrożenia	[SS1]	[SS2]	[SS3]	[SS4]	Waga	Liczba interakcji	Iloczyn wag i interakcji	Ranga
[Z1]	1	1	1	1	0,40	4	1,6	1
[Z2]	0	0	0	0	0,15	0	0	2
[Z3]	0	0	0	0	0,15	0	0	2
Waga	0,30	0,25	0,30	0,15				
Liczba interakcji	1	1	1	1		8		
Iloczyn wag i interakcji	0,3	0,25	0,3	0,15			2,6	
Ranga	1	3	1	4				

Rysunek 2.5: Czy określona słaba strona potęguje dane zagrożenie?

2.5.3 Analiza powiązań TOWS

Mocne strony/ Szanse	[MS1]	[MS2]	[MS3]	[MS4]	Waga	Liczba interakcji	Iloczyn wag i interakcji	Ranga
[S1]	1	0	0	0	0,40	1	0,4	2
[S2]	0	0	0	1	0,15	1	0,15	3
[S3]	0	0	0	1	0,15	1	0,15	3
[S4]	0	0	1	1	0,30	2	0,6	1
Waga	0,30	0,25	0,30	0,15				
Liczba interakcji	1	0	1	3		10		
Iloczyn wag i interakcji	0,3	0	0,3	0,45			2,35	
Ranga	2	3	2	1				

Rysunek 2.6: Czy określona szansa potęguje daną silną stronę?

Mocne strony/ Zagrożenia	[MS1]	[MS2]	[MS3]	[MS4]	Waga	Liczba interakcji	Iloczyn wag i interakcji	Ranga
[Z1]	1	0	0	1	0,40	2	0,8	1
[Z2]	0	0	0	0	0,40	0	0	3
[Z3]	1	0	0	0	0,20	1	0,2	2
Waga	0,30	0,25	0,30	0,15				
Liczba interakcji	2	0	0	1		6		
Iloczyn wag i interakcji	0,6	0	0	0,15			1,75	
Ranga	1	3	3	2				

Rysunek 2.7: Czy określone zagrożenie ogranicza daną silną stronę?

Słabe strony/Szanse	[SS1]	[SS2]	[SS3]	[SS4]	Waga	Liczba interakcji	Iloczyn wag i interakcji	Ranga
[S1]	0	0	0	0	0,40	0	0	3
[S2]	0	0	1	0	0,15	1	0,15	2
[S3]	1	0	0	0	0,15	1	0,15	2
[S4]	1	0	0	0	0,30	1	0,3	1
Waga	0,40	0,10	0,10	0,40				
Liczba interakcji	2	0	1	0		6		
Iloczyn wag i interakcji	0,8	0	0,1	0			1,5	
Ranga	1	3	2	3				

Rysunek 2.8: Czy określona szansa pozwala osłabić (przewyciężyć) daną słabą stronę?

Słabe strony/Zagrożenia	[SS1]	[SS2]	[SS3]	[SS4]	Waga	Liczba interakcji	Iloczyn wag i interakcji	Ranga
[Z1]	1	0	0	1	0,40	2	0,8	1
[Z2]	1	0	0	0	0,15	1	0,15	3
[Z3]	1	0	0	1	0,15	2	0,3	2
Waga	0,30	0,25	0,30	0,15				
Liczba interakcji	3	0	0	2		10		
Iloczyn wag i interakcji	0,9	0,00	0,00	0,3			2,45	
Ranga	1	3	3	2				

Rysunek 2.9: Czy określone zagrożenie wzmacnia daną słabą stronę?

2.5.4 Wybór strategii

Kombinacje	Wyniki analizy SWOT		Wyniki analizy TOWS		Zestawienie zbiorcze SWOT/TOWS	
	Suma interakcji	Suma iloczynów	Suma interakcji	Suma iloczynów	Suma interakcji	Suma iloczynów
Mocne strony/szanse	18	5,2	10	2,35	28,00	7,55
Mocne strony/zagrożenia	10	3,1	6	1,75	16,00	4,85
Słabe strony/szanse	4	1,3	6	1,5	10,00	2,8
Słabe strony/zagrożenia	8	2,6	10	2,45	18,00	5,05

Rysunek 2.10: Wyniki zbiorcze analizy SWOT/TOWS

W rozpatrywanym projekcie powinniśmy przyjąć strategię agresywną. Z zbiorczej analizy SWOT i TOWS wynik, że największą liczbę interakcji oraz sumę iloczynów ma kombinacja mocnych stron oraz szans.

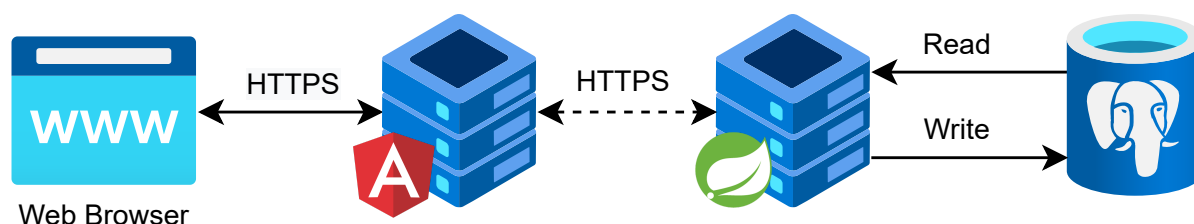
Na rynku widoczna jest rosnący popyt na mieszkania, związany m.in. z napływem dużej ilości imigrantów oraz końcem pandemii. Biorąc pod uwagę posiadany zespół specjalistów jest to niewątpliwie to szansa, aby wejść na rynek z dobrej jakości rozwiązaniem mogącym konkurować z istniejącymi rozwiązaniami. Niewątpliwym atutami jest użycie nowych technologii i zabezpieczeń wyprzedzając w ten sposób istniejące już rozwiązania. Ważnym czynnikiem jest również plan marketingowy mający na celu wypromować wprowadzaną stronę.

Rozdział 3

Projekt systemu

3.1 Architektura systemu

System będzie składał się z dwóch niezależnych aplikacji: aplikacji klienckiej, która dostarcza interfejs graficzny dla użytkownika w przeglądarce i aplikacji serwerowej odpowiadającej za komunikację z bazą danych i przetwarzanie zapytań otrzymywanych przez aplikację kliencką. Architektura systemu została przedstawiona na rysunku 3.1.



Rysunek 3.1: Architektura aplikacji trójwarstwowej

Użytkownik systemu, korzystając z przeglądarki internetowej będzie wysyłać żądania do aplikacji klienckiej, która będzie odsyłać dostępne widoki.

W momencie, w którym aplikacja kliencka będzie wymagała pobrania dodatkowych danych, zostanie wysłane żądanie *HTTP* do aplikacji serwerowej. Zapytania to będzie przetwarzane, aby następnie odesłać odpowiedź na podstawie wysłanych danych. Informacje przesyłane przez zapytania, jak i odpowiedzi od serwera będą w formacie *JSON*.

Aplikacja serwerowa z powodu posiadania dostępu do bazy danych i informacji jakie są w nich zawarte, musi w jakiś sposób zabezpieczać ten dostęp. Dlatego też dla niektórych zapytań pomiędzy klientem, a serwerem będzie wymagana autoryzacja użytkownika. Do tego zostaną wykorzystane technologie *JSON Web Tokens* wraz z *Spring Web Security*, których zadaniem będzie generowanie plików *cookies*, z informacjami o sesji użytkownika.

3.2 Technologie

- **Aplikacja frontendowa** - Angular 14.2.6 z bibliotekami Bootstrap i Material
- **Serwer** - Java 17 z frameworkiem SpringBoot w wersji 2.7.5
- **Baza Danych** - relacyjna baza danych PostgreSQL 15.1

3.2.1 Techniczna wykonalność projektu

Wszystkie informacje przechowywane będą w bazie danych PostgreSQL. Serwer komunikując się z bazą będzie się łączył przy użyciu biblioteki *Springa*, która jest *Spring Boot JPA*. Biblioteka ta zawiera w sobie framework *Hibernate*, który pozwala po uprzednim skonfigurowaniu połączenia na komunikację z warstwą danych.

W przypadku aplikacji *frontendowej* wykorzystany zostanie *Angular* z dedykowanymi bibliotekami *Bootstrap* i *Material*, których głównym zadaniem jest ułatwienie pracy nad wyglądem pisanej aplikacji.

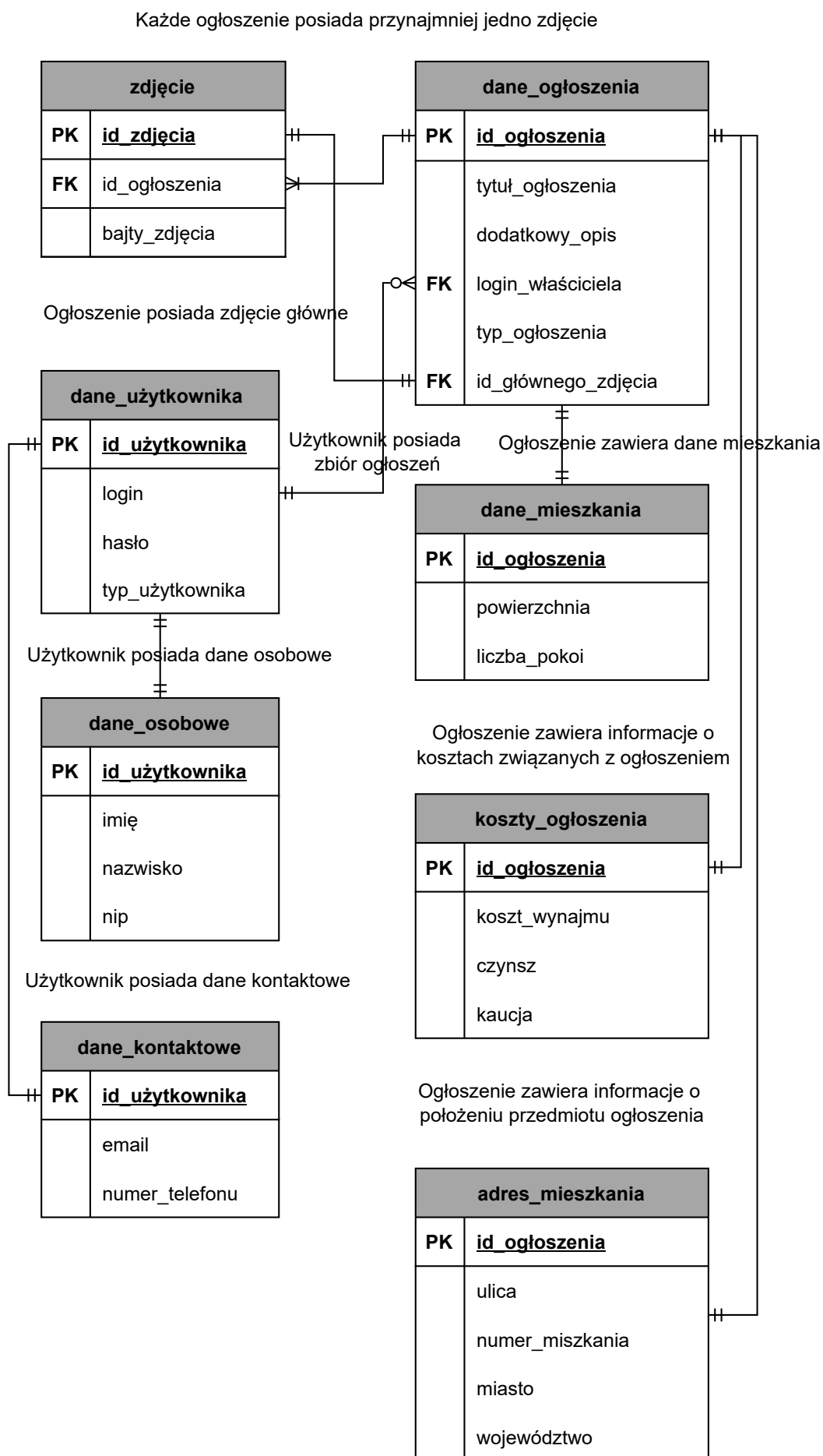
Proces wymiany informacji pomiędzy serwerem, a stroną użytkownika, będzie się odbywać z wykorzystaniem zapytań *HTTP*. Ten sposób sprawia, że nieważne jest w jakiej technologii zostaną napisane poszczególne, ponieważ i tak będą mogły się ze sobą komunikować. Dodatkowo wykorzystanie tej technologii sprawia, że strona użytkownika może być uruchomiona na każdym urządzeniu z przeglądarką internetową.

Do autoryzacji zostanie wykorzystany mechanizm *JSON Web Tokens*, którego wartość będzie przechowywana w ciasteczkach przesyłanych z serwera do klienta. Jako, że każda przeglądarka obsługuje ciasteczka mechanizm ten może zostać bezproblemowo zaimplementowany w autoryzacji konta.

3.3 Model logiczny bazy danych

Na rysunku 3.2 przedstawiono model logiczny bazy danych. Przedstawia on powiązania pomiędzy tabelami. Można wyróżnić trzy grupy tabel tj. tabelę ogłoszeniodawcy, tabelę ogłoszenia jak i tabelę zdjęć. W każdej z grup występuje relacja jeden do jednego. Dodatkowe relację to:

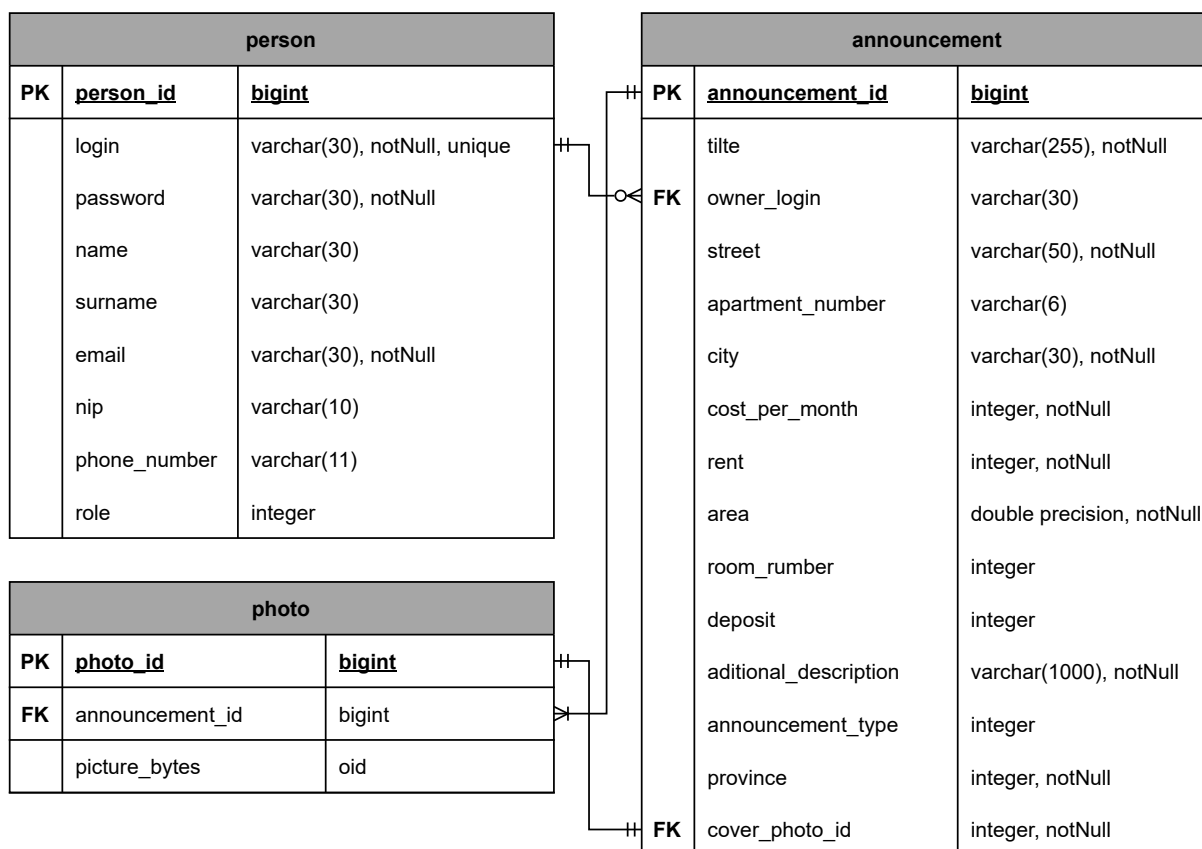
- relacja pomiędzy ogłoszeniodawcą, który może posiadać wiele ogłoszeń,
- relacja pomiędzy ogłoszeniem, a głównym zdjęciem ogłoszenia,
- relacja pomiędzy ogłoszeniem, która może posiadać wiele dodatkowych zdjęć



Rysunek 3.2: Model logiczny bazy danych

3.4 Model fizyczny bazy danych

Poniżej (rys. 3.3) załączono diagram encji wygenerowanej z wykorzystaniem *Hibernate*. Każdą tabelę z grup wymienionych w rozdziale 3.3 połączono do skompresowano do jednej tabeli. Celem tej kompresji było przyspieszenie wykonywania potencjalnie najczęściej wykonywanych operacji wyszukiwania. Baza danych przechowuje dane użytkownika, jego ogłoszenia wynajmu lub sprzedaży oraz zdjęcia wykorzystywane w ogłoszeniu. Dodatkowym zabezpieczeniem zastosowanym w bazie danych jest przechowywanie w bazie danych zaszyfrowanych haseł użytkownika.



Rysunek 3.3: Diagram encji bazy danych

Rozdział 4

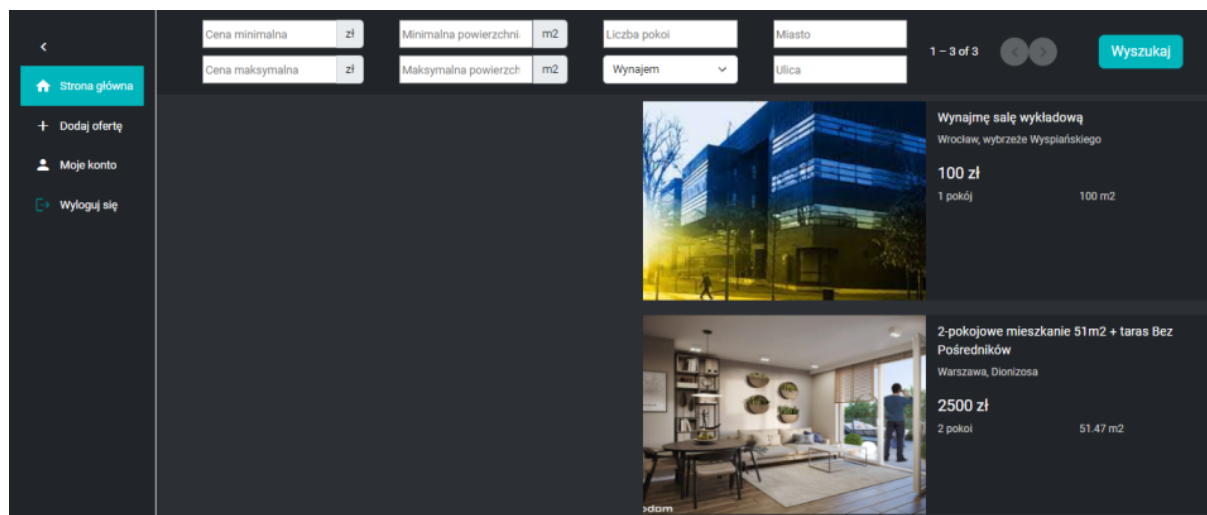
Implementacja

4.1 Perspektywa klienta

4.1.1 Widoki aplikacji

Stroną domową omawianego serwisu widoczna jest na rysunku 4.1. Składa się ona głównie z listy dostępnych w serwisie oraz filtry umożliwiające personalizację wyszukiwań. W ramach wyszukiwania skorzystać można z kilku parametrów takich jak cena, powierzchnia czy liczba pokoi. Uruchamiając stronę po raz pierwszy wyświetlane są wyniki jedynie z parametrem rodzaju ogłoszenia ustawionego na wynajem. Po naciśnięciu przycisku "Wyszukaj" pobrane zostaną oferty zgodne z podanymi filtrami oraz wyświetlone w postaci listy zawierającej przede wszystkim zdjęcie główne, tytuł oraz kilka innych informacji związanych z ogłoszeniem. W przypadku dużej liczby wyników skorzystać można z mechanizmu stronicowania, który umożliwia zobaczenie kolejnych ogłoszeń.

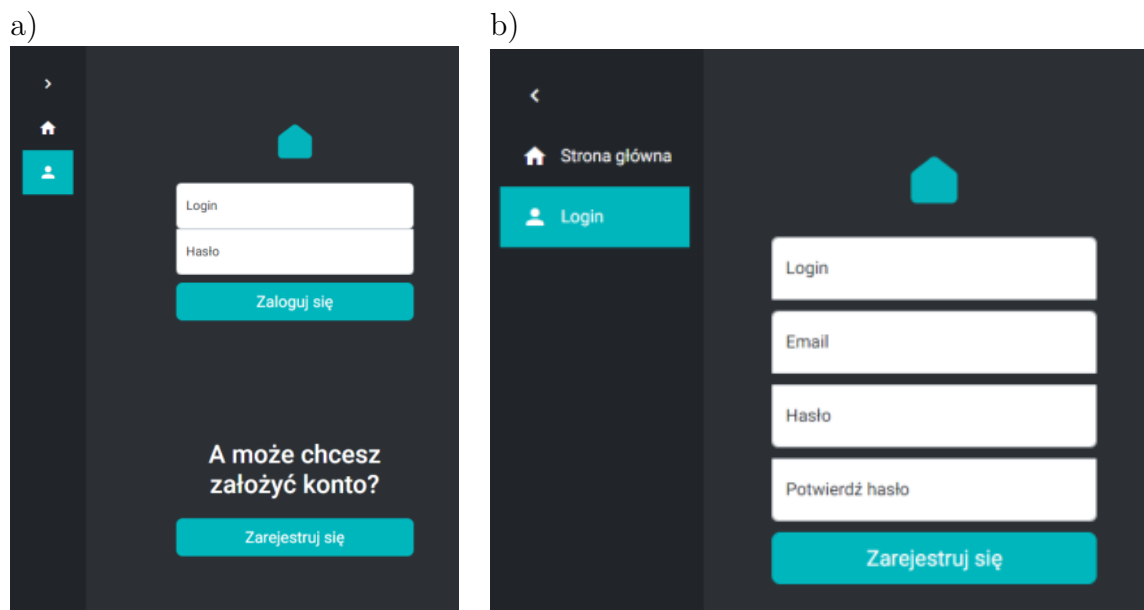
Istotnym elementem menu, który obejmuje wszystkie z widoków jest pasek nawigacji bocznej znajdujący się po lewej stronie. Dzięki niemu możemy się przełączać pomiędzy dostępnymi w serwisie akcjami. Widoczne opcje są dynamicznie zmieniane w zależności od tego czy użytkownika jest zalogowany oraz od poziomu jego dostępu. W przypadku niezalogowanego użytkownika dostępne są opcje przeglądania wszystkich ogłoszeń oraz zalogowanie i rejestracja konta. Natomiast zalogowany użytkownik może również dodawać oraz zarządzać swoimi ogłoszeniami.



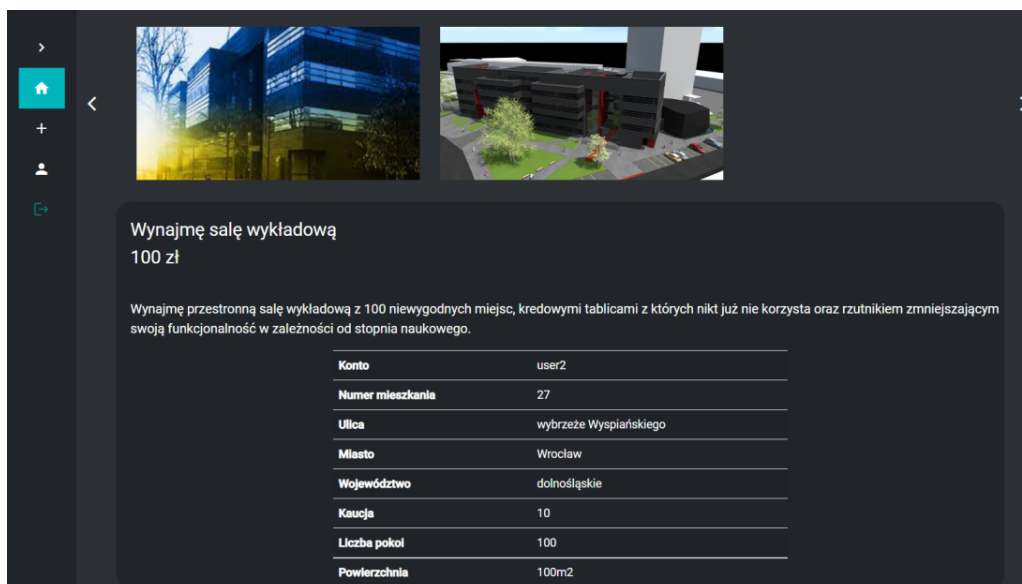
Rysunek 4.1: Główny widok aplikacji

Po naciśnięciu ogłoszenia znajdującego się na liście strona wyświetli szczegółowy widok ogłoszenia zawierający wszystkie niezbędne informacje oraz zdjęcia. Galerię zdjęć można przeglądać za pomocą widocznych na rysunku 4.3 strzałek, bądź poprzez kliknięcie wybranego ze zdjęć. Zaznaczone zdjęcie zostanie wyświetlone ponad listą zdjęć w powiększonej formie.

Ważnym elementem aplikacji jest również system logowania oraz rejestracji dostępny po kliknięciu okonki profilu w bocznej nawigacji. Wyświetlony zostanie widok przedstawiony na rysunku 4.2a. Jeżeli użytkownik chce skorzystać z istniejącego już konta to musi wypełnić pola logowania oraz nacisnąć przycisk "zaloguj się". Dla nowych użytkowników dostępna jest opcja "Zarejestruj się" po kliknięciu, której wyświetlony zostanie formularz wymagający uzupełnienia danymi. Po jego akceptacji oraz sprawdzeniu poprawności danych stworzone zostanie konto, na które można się od razu zalogować.

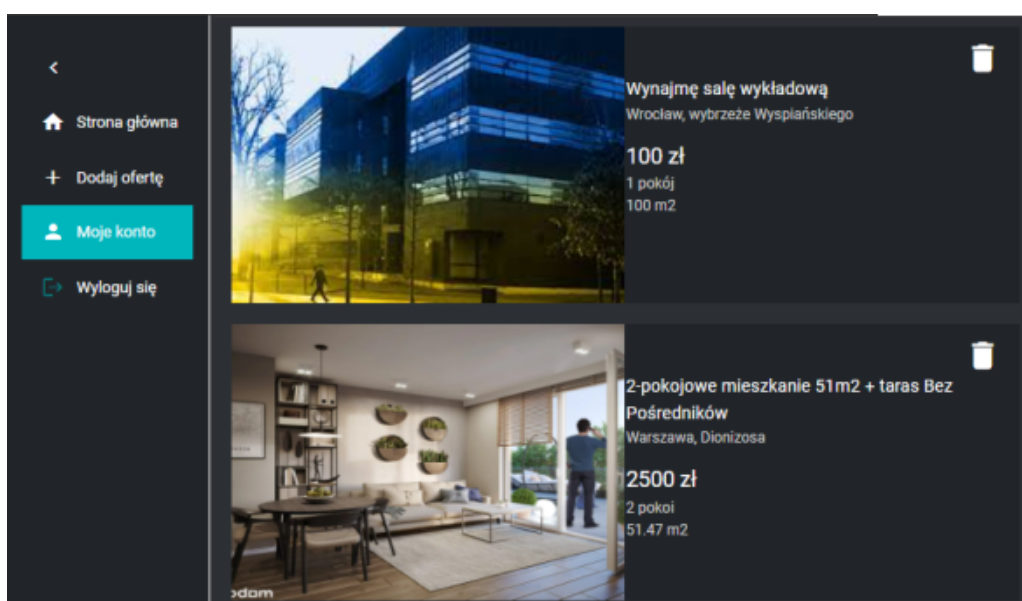


Rysunek 4.2: Widoki aplikacji: a) Logowanie, b) Rejestracja



Rysunek 4.3: Szczegóły ogłoszenia

Z perspektywy zalogowanego użytkownika dodać można nowe ogłoszenie, które przypisane zostanie do wykorzystywanego konta. Aby tego dokonać należy skorzystać z opcji "plusa" dostępnej w bocznym menu oraz uzupełnić formularz widoczny na rysunku 4.5 danymi. Aby pomyślnie sfinalizować dodanie oferty wymagana jest poprawność danych obowiązkowych oraz zamieszczenie co najmniej jednego zdjęcia. Po naciśnięciu przycisku "Dodaj ogłoszenie" strona wyświetli potwierdzenie wykonanej operacji. Od teraz ogłoszenie będzie dostępne do wglądu dla wszystkich użytkowników oraz możliwe będzie jego usunięcie przez właściciela z poziomu panelu listy ogłoszeń zalogowanego użytkownika. Wspomniany panel widoczny jest na rysunku 4.4 i zawiera listę analogiczną do tej z wszystkimi ofertami jednak wyświetlane są w niej jedynie ogłoszenia zalogowanego użytkownika i dostępna jest na nich ikona kosza.



Rysunek 4.4: Ogłoszenia użytkownika

Dodaj swoją ofertę

Dane ogłoszenia

Adres: Numer domu:

Województwo: Mięscowosc:

Rodzaj ogłoszenia

☒ Wynajem ☐ Sprzedaż

Tytuł

Opis ogłoszenia

Cena **Czynsz**

Cena w PLN

Powierzchnia **Liczba pokoi**

Powierzchnia w m2

Dodaj ogłoszenie

Zdjęcia

Dodaj zdjęcia

Rysunek 4.5: Dodanie ogłoszenia

4.2 Panel administratora

W sytuacji gdy konto zalogowanego użytkownika posiada uprawnienia administratora to w menu dostępna będzie również dodatkowa akcja pod nazwą "Panel administratora". Umożliwia ona przegląd oraz zarządzanie użytkownikami istniejącymi w systemie.

Panel administratora

Użytkownicy:

- Imię:** Robert
Nazwisko: Lewandowski
Login: robecik
Email: robecik@wp.pl
NIP: 1234563218
Numer telefonu: 680543260
- Imię:** Wojtek
Nazwisko: Glazer
Login: wojteczek
Email: wojteczek@gmail.com
NIP: 5234564218
Numer telefonu: 280543260
- Imię:** Kamil
Nazwisko: Kowalski
Login: kamilBezUcha
Email: kamilBezUcha@gmail.com
NIP: 5234523218
Numer telefonu: 7805432260

Rysunek 4.6: Usuwanie użytkowników

4.3 Perspektywa programisty

4.3.1 Punkty dostępu do serwera

Punkty dostępu do serwera to interfejs, do który pozwala na interakcję z serwerem. W serwerze wyszczególniono trzy węzły komunikacyjne:

- *AnnouncementController* - punkt dostępu dla nie zalogowanych użytkowników (Najemców), daje dostęp do przeglądania ogłoszeń,
- *AuthenticatedControllers* - punkt dostępu dla zalogowanych użytkowników (Wynajmujących), pozwala na dodawanie ogłoszeń,
- *AuthorizationControllers* - punkt dostępu wykorzystywany do logowania i rejestrowania się użytkowników.

4.3.2 Przykładowa implementacja punktu dostępu

W listingu 4.1 przedstawiono przykładowy punkt dostępu, który dodaje przekazane jako parametr ogłoszenie do bazy danych. W przypadku pozytywnego wykonania operacji wysyłana jest informacja o dodaniu ogłoszenia wraz z przydzielonym indeksem. Jeżeli ogłoszenie nie zostało dodane przekazywana jest informacja o nie dodaniu ogłoszenia.

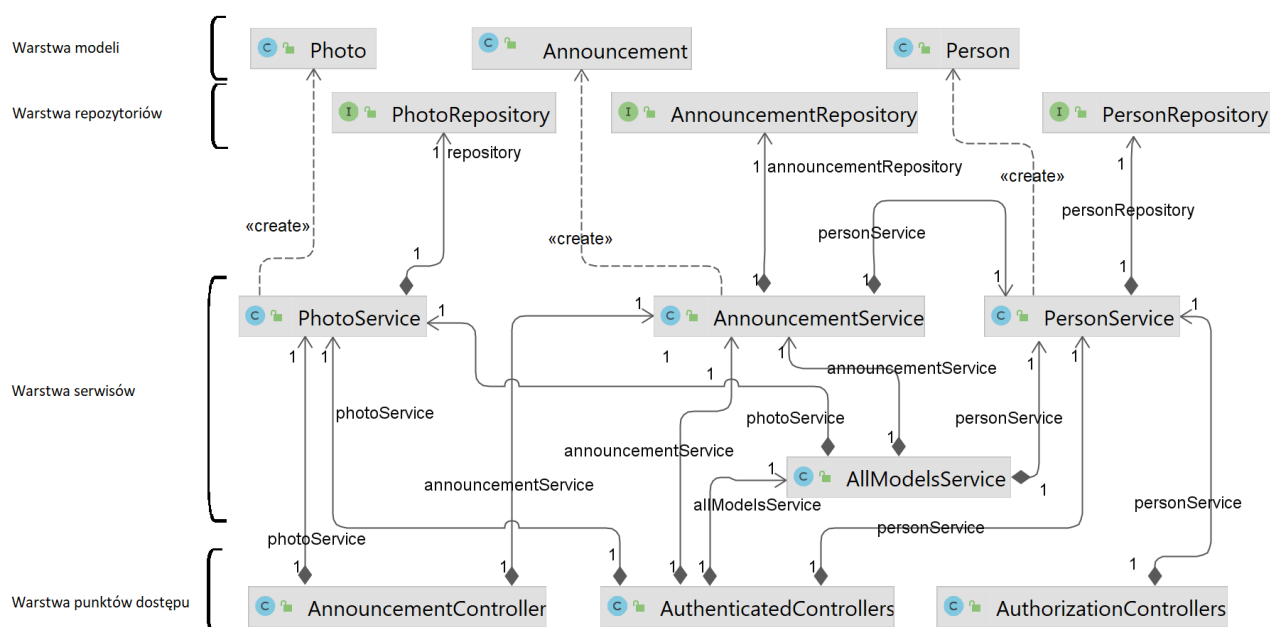
```
@PostMapping("/addAnnouncement")
public ResponseEntity<?> createAnnouncement(@RequestBody
    Announcement announcementRequest) {
    try {
        Optional<Announcement> announcement =
            announcementService.createNewAnnouncement(announcementRequest);
        return announcement.<ResponseEntity<?>>map(value ->
            ResponseEntity.ok(
                new ResponseJsonBody("Create announcement number: "
                    + value.getAnnouncementId()))
            .orElseGet(() -> ResponseEntity.badRequest().body(
                new ResponseJsonBody("No announcement created")));
    } catch (Exception e) {
        return ResponseEntity.accepted().body(new
            ResponseJsonBody(e.getMessage()));
    }
}
```

Listing 4.1: Punkt dostępu do dodawania ogłoszeń

4.3.3 Schemat połączenia punktów dostępu i bazy danych

Na poniższym (rys. 4.7) przedstawiono warstwową strukturę klas wykorzystywaną do komunikacji pomiędzy punktami dostępu, a bazą danych. Kolejne warstwy to:

- warstwa modeli zawierająca klasy odpowiadające tabelą przedstawionym na diagramie 3.3,
- warstwa repozytoriów zawierająca klasy z metodami implementującymi operacje odczytu i zapisu z bazy danych,
- warstwa serwisów zawierająca klasy z metodami operującymi na różnych metodach warstwy repozytoriów,
- warstwa punktów dostępu zawierająca klasy przetwarzające zapytania.



Rysunek 4.7: Diagram kluczowych klas w dostępie do danych bazy danych

4.3.4 Zabezpieczenie dostępu do endpointów

W celu zabezpieczenia informacji przechowywanych na serwerze wprowadzono autoryzację użytkowników. Autoryzacja odbywa się przy wykorzystaniu biblioteki *Spring Boot Starter Security* i *JSON Web Token*. JWT odpowiada za generowanie tokenów, które następnie są sprawdzane aby można było stwierdzić czy dany użytkownik przechodzi proces autoryzacji. Fragment kodu odpowiadającego za walidację tokenu i danych w nim zawartych znajduje się w listingu 4.2.

```
cookieManager.getJwtFromCookies(request)
    .flatMap(cookieManager::checkJwtValidation)
    .ifPresent(login -> {
try {
    UserDetails userDetails =
        userDetailsService.loadUserByUsername(login);
    UsernamePasswordAuthenticationToken authentication =
        new
            UsernamePasswordAuthenticationToken(userDetails,
            userDetails.getPassword(),
            userDetails.getAuthorities());

    authentication.setDetails(
        new WebAuthenticationDetailsSource()
            .buildDetails(request));

    SecurityContextHolder.getContext()
        .setAuthentication(authentication);
} catch (UsernameNotFoundException e) {
    LOGGER.info("This login doesn't exist: {} ",
        e.getMessage());
}
});
```

Listing 4.2: Metoda odpowiadająca za sprawdzenie danych logowania w ciasteczku

Cały proces udzielenia dostępu do zabezpieczonego *endpointu* zaczyna się od sprawdzenia adresu, z którego pochodzi zapytanie, jeśli nie jest on adresem serwera *frontendowego* jest on automatycznie odrzucany. Następnie z przesłanego ciasteczka pobierany jest JWT, jeśli token jest ważny i prawidłowy to wyciągane są z niego dane potrzebne do uwierzytelnienia użytkownika. W wypadku gdy dane logowania są poprawne to klient uzyskuje dostęp do danych.

4.3.5 Generowanie ciasteczka dla zalogowanego użytkownika

Wraz z żądaniem o zalogowanie się, sprawdzana jest poprawność loginu i hasła użytkownika. W sytuacji gdy dane są poprawne generowany jest token JWT, który zawiera takie informacje jak login czy daty ważności tokenu (listing 4.6).

```
private String createJsonWebToken(String login) {  
    return JWT.create()  
        .withSubject(login)  
        .withIssuedAt(new Date())  
        .withExpiresAt(new Date(new Date().getTime() +  
            expirationTime))  
        .sign(algorithm);  
}
```

Listing 4.3: Tworzenie JWT na podstawie loginu

Następnie wygenerowany token jest umieszczany w ciasteczku, w którym dodatkowo ustalany jest czas wygaśnięcia na 30 minut. Aby ograniczyć sytuację, w której użytkownik zapomniał się wylogować ze swojego konta. Fragment kodu znajduje się w listingu 4.4.

```
public ResponseCookie generateCookieWithJwt(String login) {  
    String jwtToken = createJsonWebToken(login);  
    return ResponseCookie  
        .from(cookieName, jwtToken)  
        .path(AUTHENTICATED_ENDPOINT)  
        .maxAge(expirationTime / 1000)  
        .httpOnly(true)  
        .build();  
}
```

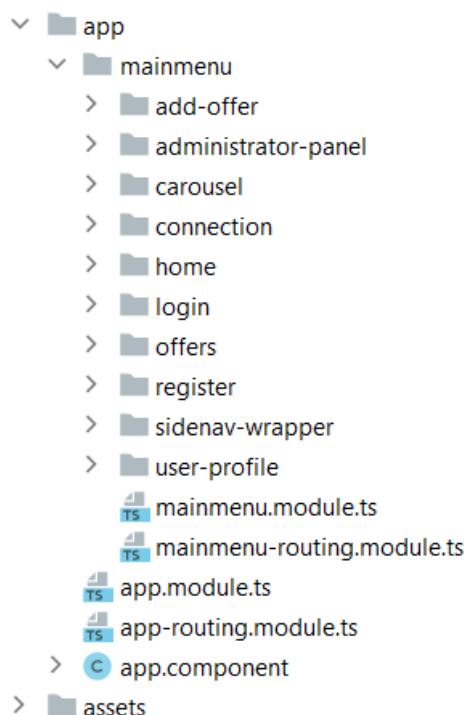
Listing 4.4: Tworzenie ciasteczka zawierającego JSON Web Token

4.3.6 Struktura projektu strony użytkownika

Każdy z widoków wymienionych w punkcie 4.1.1 został zaimplementowany w osobnym komponencie (4.8). W skład komponentu wchodzi:

- plik *HTML* opisujący ułożenie elementów na stronie,
- plik *ts* implementujący logikę strony,
- plik *CSS* opisujący wygląd elementów na stronie.

Niektóre katalogi zawierają plik formularza wykorzystywanego na danej stronie. Formularz definiuje obiekt wysyłany do serwera, W Formularzu opisane są dostępne pola wraz z ich typami oraz wymaganiami zawartości. Jeżeli dane wymaganie nie jest spełnione nie ma możliwości wysłania obiektu w zapytaniu. Dodatkowo wyświetlana jest dodatkowa



Rysunek 4.8: Struktura projektu strony użytkownika

informacja co zostało źle wprowadzone. Przykład implementacji został umieszczony na rysunku 4.5

```

<div *ngIf="announcementSubmitted && f.street.errors"
  class="invalid-feedback">
  <div *ngIf="f.street.errors.required">Nazwa ulicy jest
    wymagana</div>
</div>

```

Listing 4.5: Walidacja wprowadzonych danych

Zapytania wysyłane do części backendowej aplikacji realizowane są w specjalnie wydzielonym serwisie w celu oddzielenia logiki obsługi widoków od asynchronicznych zapytań do API. Podczas ładowania widoku lub naciśnięcia przycisku wysyłane jest zapytanie do backendu. W celu otrzymania wyników zapytania komponent obsługujący widok subskrybuje dane z serwisu, dzięki czemu otrzymuje on informacje o uzyskaniu odpowiedzi od serwera. Przykładowe zapytanie o listę ofert widoczne jest na poniższym listingu.

```

this.http
  .get<Offer[]>(
    'http://localhost:8080/public/announcements/', {
      params: params } )
  .pipe(
    map(recipes => { return recipes

```

```
.map(recipe => {  
    return { ...recipe, }; });  
}),  
tap(recipes => {  
    console.log(recipes.length)  
    this.offers = recipes  
    this.offersChanged.next(recipes.slice());  
})  
) .subscribe()
```

Listing 4.6: Wysyłanie zapytania o listę ofert

4.4 Perspektywa wdrożeniowca

System do swojego działania potrzebuje w pełni działających trzech warstw:

- prezentacji,
- serwera,
- danych.

Pierwszym krokiem jest skonfigurowanie bazy danych, jej kont dostępowych oraz umieszczenie jej na serwerze.

W kolejnym kroku należy skonfigurować część *backendową*. Aplikacja serwerowa do swojej wersji produkcyjnej potrzebuje zdefiniowania w plikach ustawień dwóch specyfikacji, gdzie pierwszym z nich jest adres aplikacji *frontendowej*, który umożliwia dostępu do *endpointów* serwera oraz dane potrzebne do nawiązania połączenia z bazą danych takie jak hasło, nazwa użytkownika czy adres. Dodatkowo można zmienić ustawienia ciasteczek wykorzystywanych przy autoryzacji użytkowników korzystających z systemu. Po skonfigurowaniu warstwy serwera należy wywołać polecenie *mvn install* generujące wersję *release* pliku *jar*. Takiego wygenerowanego *jara* można umieścić na serwerze.

Aplikacja *frontendowa* do swojej konfiguracji i wdrożenia potrzebuje skonfigurować adres pod jakim znajduje się aplikacja serwerowa, żeby mogła pobierać dane przetwarzane przez aplikację *backendową*. Następnie wymagane jest wywołanie polecenia *ng build*, które tworzy wersję dystrybucyjną, gotową do instalacji.

Rozdział 5

Podsumowanie

W ramach projektu udało się zrealizować założone cele. Aplikacja umożliwia zrealizowanie przypadków użycia zawartych w planie projektu. Dostępne są funkcjonalności związane zarówno z procesem logowania i rejestracji użytkowników jak i logiką obsługi ofert - w tym ich dodawanie, usuwanie oraz wyświetlanie.

Przy implementacji wykorzystane zostały technologie wymienione w części poświęconej projektowi systemu. Ich dobór umożliwił wytworzenie trójwarstwowej aplikacji, w której komunikacja odbywa się za pomocą mechanizmów zapewniających bezpieczeństwo dla użytkowników systemu.

Finalny produkt powinien zostać wdrożony przy użyciu strategii agresywnej. Wybór strategii został poparty analizą SWOT oraz TOWS zawartej w rozdziale poświęconym analizie wymagań.