



# POLSKO-JAPONSKA AKADEMIA TECHNIK KOMPUTEROWYCH

**Wydział Informatyki**  
**Filia w Gdańsku**

**Langmesser Adam**

Nr albumu s27119

Nazwa specjalizacji: Aplikacje Internetowe

**Redosz Mateusz**

Nr albumu s27094

Nazwa specjalizacji: Aplikacje Internetowe

**Oziemczuk Stanisław**

Nr albumu s26982

Nazwa specjalizacji: Aplikacje Internetowe

**Badek Kacper**

Nr albumu s29168

Nazwa specjalizacji: Aplikacje Internetowe

## Aplikacja webowa: [spotty-na-drony.pl](http://spotty-na-drony.pl)

Rodzaj pracy  
inżynierska  
Imię i nazwisko promotora  
mgr Adam Urbanowicz

Gdańsk, miesiąc, 2100 obrony

**Streszczenie:** Celem niniejszej pracy było stworzenie w pełni funkcjonalnej i działającej aplikacji internetowej pozwalającej na szybkie wyszukiwanie spotów w okolicy oraz dzielenie się zdjęciami, filmami oraz doświadczeniem z innymi użytkownikami. W ramach pracy stworzono system składający się z trzech komponentów: [Frontendu](#), [Backendu](#) oraz bazy-danych. Aplikacja internetowa została wykonana przy pomocy [Frameworka](#) React w językach Javascript oraz Typescript, do stylów został użyty Tailwind. Serwis backendowy został stworzony w języku Java oraz biblioteki Spring Boot. Baza danych to PostgreSql.

Komunikacja między komponentami odbywała się zgodnie ze standardem REST. Projekt został zrealizowany w podejściu ewolucyjno-przyrostowym z elementami Kanban.

**Słowa kluczowe:** — brak —



# POLSKO-JAPONSKA AKADEMIA TECHNIK KOMPUTEROWYCH

## Karta projektu

<b>Temat projektu:</b> Aplikacja webowa: spotty-na-drony.pl <b>Temat projektu po angielsku:</b> Web application: spotty-na-drony.pl	<b>Akronim:</b> Merkury <b>Data ustalenia tematu</b> 2023-10-10
<b>Promotor:</b>  mgr Adam Urbanowicz	<b>Konsultanci:</b>  1. — brak —
<b>Cele projektu:</b> Stworzenie w pełni funkcjonalnej aplikacji internetowej do rozwijania hobby(latania dronem).	
<b>Rezultaty projektu:</b> Aplikacja Internetowa, Dokumentacja Interaktywna mapa z wyświetlanymi spotami oraz pogodą. Zaawansowana wyszukiwarka spotów. Forum do dzielenia się informacjami na temat dronów. Chat jednoosobowy oraz grupowy. Konto użytkownika z możliwością zapisania ulubionych spotów.	
<b>Miary sukcesu:</b> Gotowa do wdrożenia aplikacja. Realizacja w terminie zgodnym z wymaganiami.	
<b>Ograniczenia:</b> Budżetowe: brak środków na wdrożenie. Zawodowe: brak doświadczenia. Czasowe: trzy semestry (09.2024 - 02.2026). Ludzkie: czteroosobowy zespół.	

Wykonawcy	Numer albumu	Specjalizacja	Tryb studiów
Langmesser Adam	s27119	Aplikacje Internetowe	Stacjonarny
Redosz Mateusz	s27094	Aplikacje Internetowe	Stacjonarny
Oziemczuk Stanisław	s26982	Aplikacje Internetowe	Stacjonarny
Badek Kacper	s29168	Aplikacje Internetowe	Stacjonarny

<b>Data ukończenia projektu:</b> 7 grudnia 2025	<b>Recenzent:</b> dr Elżbieta Puźniakowska-Gałuch
--	--

# Spis treści

<b>1 Wstęp</b>	<b>7</b>
1.1 O projekcie . . . . .	7
1.2 Cel i zakres prac . . . . .	7
1.3 Geneza pomysłu . . . . .	7
<b>2 Opis problemu</b>	<b>8</b>
2.1 Rich picture . . . . .	8
2.2 Udziałowcy . . . . .	8
2.3 Istniejące rozwiązania . . . . .	10
2.4 Wizja rozwiązania . . . . .	10
2.5 Aspekty społeczne i biznesowe . . . . .	10
2.5.1 Aspekty społeczne . . . . .	10
2.5.2 Aspekty biznesowe . . . . .	10
<b>3 Planowanie</b>	<b>11</b>
3.1 Metodologia pracy . . . . .	11
3.1.1 Przegląd rozważanych podejść . . . . .	11
3.1.2 Odrzucone podejścia . . . . .	11
3.1.3 Wybrane podejście: Disciplined Agile Delivery (Lean Life Cycle) . . . . .	12
3.1.4 Narzędzia i komunikacja . . . . .	12
3.1.5 Podział ról w zespole . . . . .	13
3.2 Harmonogram projektu . . . . .	13
3.3 Technologie i narzędzia . . . . .	15
3.3.1 Technologie . . . . .	15

3.3.2	Narzędzia . . . . .	15
3.4	Zasoby i ograniczenia . . . . .	18
3.4.1	Zasoby . . . . .	18
3.4.2	Ograniczenia . . . . .	18
3.4.3	Usługi zewnętrzne . . . . .	18
3.5	Analiza ryzyka . . . . .	21
<b>4</b>	<b>Analiza wymagań</b>	<b>22</b>
4.1	Przypadki użycia . . . . .	23
4.1.1	Aktorzy . . . . .	23
4.1.2	Diagramy przypadków użycia . . . . .	25
4.1.3	Scenariusze przypadków użycia . . . . .	33
4.1.3.1	Scenariusze przypadków użycia – funkcje ogólne .	33
4.1.3.2	Scenariusze przypadków użycia dla funkcji premium	40
4.1.3.3	Scenariusze przypadków użycia dla wyszukiwarki .	43
4.1.3.4	Scenariusze przypadków użycia dla mapy . . . . .	45
4.1.3.5	Scenariusze przypadków użycia dla czatu . . . . .	49
4.1.3.6	Scenariusze przypadków użycia dla forum . . . . .	57
4.1.3.7	Scenariusze przypadków użycia dla panelu użytkownika . . . . .	65
4.2	Wymagania . . . . .	74
4.2.1	Wymagania ogólne . . . . .	75
4.2.1.1	Wymagania ogólne dla czatu . . . . .	75
4.2.2	Wymagania funkcjonalne . . . . .	77
4.2.2.1	Funkcjonalności dla mapy . . . . .	77
4.2.2.2	Wymagania funkcjonalne dla czatu . . . . .	77
4.2.2.3	Funkcjonalności dla forum . . . . .	89
4.2.2.4	Funkcjonalności dla konta użytkownika . . . . .	89
4.2.2.5	Funkcjonalności dla logowania i rejestracji . . . . .	99
4.2.2.6	Funkcjonalności dla wyszukiwarki spotów . . . . .	100
4.2.2.7	Funkcjonalności dla motywów . . . . .	102
4.2.3	Wymagania pozafunkcjonalne . . . . .	104

4.2.3.1	Wymagania pozafunkcjonalne dla czatu . . . . .	104
<b>5</b>	<b>Projekt</b>	<b>115</b>
5.1	Wzorce projektowe . . . . .	115
5.2	Architektura systemu . . . . .	115
5.2.1	Diagram architektury . . . . .	116
5.2.2	Komponenty systemu . . . . .	118
5.3	Projekt bazy danych . . . . .	119
5.3.1	Model danych . . . . .	119
5.3.2	Diagram ERD . . . . .	119
5.4	Architektura interfejsu użytkownika . . . . .	119
5.4.1	Projekt strony głównej . . . . .	119
5.4.2	Projekt panelu logowania . . . . .	119
5.4.3	Projekt mapy . . . . .	119
5.4.4	Projekt chatu . . . . .	119
5.4.5	Projekt forum . . . . .	119
5.4.6	Projekt konta użytkownika . . . . .	119
<b>6</b>	<b>Przebieg realizacji projektu</b>	<b>120</b>
6.1	Sprint 1 . . . . .	120
6.2	Sprint 2 . . . . .	120
<b>7</b>	<b>Realizacja Projektu</b>	<b>121</b>
7.1	Implementacja backendu . . . . .	121
7.1.1	Struktura projektu . . . . .	121
7.1.2	Endpointy systemu . . . . .	124
7.1.3	Integracja z bazą danych . . . . .	170
7.1.4	Obsługa uwierzytelnienia . . . . .	173
7.1.5	Konteneryzacja . . . . .	173
7.2	Implementacja frontendu . . . . .	173
7.2.1	Struktura aplikacji . . . . .	173
7.2.2	Zarządzanie stanem i przepływ danych . . . . .	178
7.2.3	Integracja i komunikacja z backendem . . . . .	181

7.2.4	Style . . . . .	184
7.2.5	Wyszukiwarka spotów . . . . .	188
7.2.6	Mapa . . . . .	195
7.2.7	Chat . . . . .	195
7.2.8	Forum . . . . .	195
7.2.9	Konto użytkownika . . . . .	195
7.2.10	Panel logowania . . . . .	195
7.3	Implementacja CI/CD . . . . .	195
<b>8</b>	<b>Testy</b>	<b>196</b>
8.1	Testy jednostkowe . . . . .	196
8.2	Testy integracyjne . . . . .	196
8.3	Testy E2E . . . . .	196
8.4	Wyniki testów i wnioski . . . . .	196
<b>9</b>	<b>Prezentacja systemu</b>	<b>197</b>
9.1	Strona główna . . . . .	197
9.2	Strona mapy . . . . .	197
9.3	Strona chatu . . . . .	197
9.4	Strona forum . . . . .	197
9.5	Panel logowania . . . . .	197
9.6	Panel konta użytkownika . . . . .	197
<b>10</b>	<b>Nakład pracy</b>	<b>198</b>
10.1	Ogólny nakład pracy . . . . .	198
10.2	Indywidualne nakłady pracy . . . . .	198
10.2.1	Adam Langmesser . . . . .	198
10.2.2	Mateusz Redosz . . . . .	198
10.2.3	Stanisław Oziemczuk . . . . .	201
10.2.4	Kacper Badek . . . . .	201
<b>11</b>	<b>Podsumowanie</b>	<b>202</b>
11.1	Osiągnięte rezultaty . . . . .	202
11.2	Napotkane wyzwania . . . . .	202

11.3 Plany na przyszłość . . . . .	202
<b>12 Słownik pojęć i skrótów</b>	<b>203</b>
<b>Spis tabel</b>	<b>211</b>
<b>Bibliografia</b>	<b>218</b>
<b>Załączniki</b>	<b>220</b>

# Rozdział 1

## Wstęp

1.1 O projekcie

1.2 Cel i zakres prac

1.3 Geneza pomysłu

## Rozdział 2

### Opis problemu

#### 2.1 Rich picture

#### 2.2 Udziałowcy

KARTA UDZIAŁOWCA	
<b>Identyfikator:</b>	UO1
<b>Nazwa udziałowca:</b>	Zespół projektowy
<b>Opis:</b>	Zespół czterech studentów odpowiedzialnych za analizę, projekt, implementację, testy oraz dokumentację systemu.
<b>Typ:</b>	ożywiony, bezpośredni
<b>Perspektywa:</b>	Techniczna, wykonawcza.
<b>Ograniczenia:</b>	Ograniczone zasoby czasowe i doświadczenie komercyjne.
<b>Powiązane wymagania:</b>	Wymagania funkcjonalne i techniczne systemu, możliwość realizacji w ramach projektu dyplomowego.

**Tabela 2.1:** Karta udziałowca: Zespół projektowy

KARTA UDZIAŁOWCA	
<b>Identyfikator:</b>	UO2
<b>Nazwa udziałowca:</b>	Promotor
<b>Opis:</b>	Osoba nadzorująca przebieg projektu, weryfikująca poprawność merytoryczną i zgodność z wymaganiami uczelni.
<b>Typ:</b>	ożywiony, pośredni
<b>Perspektywa:</b>	Merytoryczna, formalna, jakościowa.
<b>Ograniczenia:</b>	Nie odpowiada za implementację; rekomenduje, opiniuje i załatwia.
<b>Powiązane wymagania:</b>	Czytelna dokumentacja, zgodność z wytycznymi kierunku oraz odpowiedni poziom techniczny rozwiązania.

**Tabela 2.2:** Karta udziałowca: Promotor

KARTA UDZIAŁOWCA	
<b>Identyfikator:</b>	UO3
<b>Nazwa udziałowca:</b>	Droniarze
<b>Opis:</b>	Główna grupa docelowa systemu – osoby latające dronami rekreacyjnie lub półprofesjonalnie, szukające miejsc do lotów i wymiany doświadczeń.
<b>Typ:</b>	ożywiony, bezpośredni
<b>Perspektywa:</b>	Użytkownik końcowy: prostota obsługi, rzetelne informacje o spotach, wygodne dzielenie się treściami.
<b>Ograniczenia:</b>	Brak wpływu na architekturę techniczną systemu; oczekują intuicyjnego interfejsu.

<b>Powiązane wymagania:</b>	Lista spotów, informacje o ograniczeniach prawnych, oceny i komentarze, dodawanie treści oraz podstawowe funkcje społecznościowe.
-----------------------------	---

**Tabela 2.3:** Karta udziałowca: [Droniarze](#)

## 2.3 Istniejące rozwiązania

## 2.4 Wizja rozwiązania

## 2.5 Aspekty społeczne i biznesowe

### 2.5.1 Aspekty społeczne

### 2.5.2 Aspekty biznesowe

# Rozdział 3

## Planowanie

### 3.1 Metodologia pracy

#### 3.1.1 Przegląd rozważanych podejść

Przy wyborze metodologii pracy rozważono trzy podejścia do prowadzenia projektu informatycznego:

- klasyczny Agile (w praktyce: Scrum),
- model kaskadowy (Waterfall),
- [Disciplined Agile Delivery - Lean Life Cycle](#).

#### 3.1.2 Odrzucone podejścia

„Klasyczny Agile” (Scrum).

Mimo elastyczności i popularności zakłada pracę w iteracjach 2–4 tygodni oraz stały zestaw ceremonii (planowanie, przegląd, retrospektyna). Ze względu na nierównomierną dostępność zasobów w kolejnych miesiącach studiów nie zapewniono możliwości utrzymania stałej kadencji sprintów, dlatego z podejścia zrezygnowano.

**Model kaskadowy (Waterfall).**

Przewiduje sekwencyjne przechodzenie przez z góry określone etapy i ogranicza bieżącą weryfikację wymagań w trakcie prac deweloperskich. W projekcie wyma-

gano możliwości częstych rewizji założeń oraz wprowadzania istotnych zmian w docelowej wizji rozwiązania; dlatego z podejścia zrezygnowano.

### 3.1.3 Wybrane podejście: Disciplined Agile Delivery (Lean Life Cycle)

Podjęto decyzję o zastosowaniu **Disciplined Agile Delivery** [1] w wariantie **Lean Life Cycle** [2], ponieważ podejście to łączy pożądane cechy Agile i Waterfall, a jednocześnie eliminuje stałe sprinty na rzecz pracy w ciągłym przepływie.

**Kluczowe argumenty wyboru:**

- **Brak sprintów.** Zastosowano przepływ ciągły, co pozwala dopasować tempo do zmiennej dostępności zespołu i unikać sztucznego „domykania” iteracji.
- **Rozbudowana faza startowa.** Na początku przewidziano większy wysiłek planistyczny: doprecyzowanie zakresu, wstępna wizja architektury, identyfikacja ryzyk, plan publikacji oraz kryteria jakości – bez zamrażania szczegółów.
- **Ciągła weryfikacja wymagań.** W trakcie realizacji przewidziano bieżące doprecyzowywanie backlogu, regularny feedback promotoru oraz możliwość korygowania kierunku bez kosztów „przeskakiwania” między fazami.
- **Praktyki Lean i koncentracja na wartości.** Priorytetyzacja wartości biznesowej, wizualizacja pracy, małe partie dostaw.
- **Lekka governance i kamienie milowe.** Zastosowano lekkie mechanizmy nadzoru (peer review, prezentacje postępów) zapewniające przejrzystość bez nadmiernej biurokracji.

### 3.1.4 Narzędzia i komunikacja

Do zarządzania zadaniami zastosowana została **Jira** (monitorowanie postępu prac oraz ewidencja zadań członków zespołu). Komunikację w zespole zaplanowano w formie regularnych spotkań oraz asynchronicznie z wykorzystaniem **Discordu** oraz **Messengera**.

### **3.1.5 Podział ról w zespole**

- Adam - fullstack developer, lider zespołu
- Stanisław - fullstack developer
- Kacper - fullstack developer
- Mateusz - fullstack developer

Każdy z członków zespołu uczestniczy również w przygotowaniu dokumentacji.

## **3.2 Harmonogram projektu**

W poniższym harmonogramie przedstawiono plan prac nad poszczególnymi częściami projektu, rozłożony na miesiące.

### **Rok 2024**

**Czerwiec**     • Zebranie zespołu.

- Rozważenie potencjalnych pomysłów.

**Lipiec**     • Wybór technologii.

- Wstępne założenia architektoniczne.

**Sierpień**     • *(do uzupełnienia)*

- *(do uzupełnienia)*

**Wrzesień**     • *(do uzupełnienia)*

- *(do uzupełnienia)*

**Październik**     • *(do uzupełnienia)*

- *(do uzupełnienia)*

**Listopad**     • *(do uzupełnienia)*

- *(do uzupełnienia)*

**Grudzień** • (do uzupełnienia)

- (do uzupełnienia)

## Rok 2025

**Styczeń** • (do uzupełnienia)

- (do uzupełnienia)

**Luty** • (do uzupełnienia)

- (do uzupełnienia)

**Marzec** • (do uzupełnienia)

- (do uzupełnienia)

**Kwiecień** • (do uzupełnienia)

- (do uzupełnienia)

**Maj** • (do uzupełnienia)

- (do uzupełnienia)

**Czerwiec** • (do uzupełnienia)

- (do uzupełnienia)

**Lipiec** • (do uzupełnienia)

- (do uzupełnienia)

**Sierpień** • (do uzupełnienia)

- (do uzupełnienia)

**Wrzesień** • (do uzupełnienia)

- (do uzupełnienia)

**Październik** • (do uzupełnienia)

- (do uzupełnienia)

**Listopad** • (*do uzupełnienia*)

- (*do uzupełnienia*)

**Grudzień** • (*do uzupełnienia*)

- (*do uzupełnienia*)

## Rok 2026

**Styczeń** • (*do uzupełnienia*)

- (*do uzupełnienia*)

### 3.3 Technologie i narzędzia

Do realizacji projektu wykorzystano wiele technologii oraz narzędzi informatycznych. Przy wyborze technologii kierowaliśmy się ich popularnością, dostępnością dokumentacji oraz artykułów, a także łatwością użycia. Narzędzia zostały dopasowane do wybranych technologii i specyfikacji zadań. Poniżej przedstawiono opis wybranych opcji.

#### 3.3.1 Technologie

#### 3.3.2 Narzędzia

Do niektórych płatnych narzędzi otrzymano bezpłatny dostęp za pośrednictwem uczelni, w innych istniała możliwość założenia konta edukacyjnego, które oferowało dostęp do wszystkich funkcji narzędzia. Gdy żadna z wymienionych opcji nie była udostępniona, wybierano rozwiązania darmowe.

##### • IntelliJ IDEA Ultimate

Jest to [IDE](#) od firmy JetBrains. Dzięki licznie dostępnym pluginom oferuje obsługę wielu języków programowania oraz innych składni. Pozwala również na integrację z repozytorium. Używano go do programowania zarówno [frontendu](#), jak i [backendu](#) oraz tworzenia dokumentacji w LaTeX.

- **Docker Desktop**

To narzędzie do zarządzania obrazami, kontenerami oraz wolumenami Docker. Zawiera w sobie również silnik tej technologii. Wykorzystywano je do lokalnego uruchamiania bazy danych oraz serwisu do cachowania.

- **One Drive**

Usługa dysku chmurowego oferowana przez firmę Microsoft. Przechowywano tam dokumenty oraz obrazy diagramów.

- **Azure Blob Storage**

To rozwiązanie chmurowe Microsoft, służące do bezpiecznego przechowywania dużej ilości danych niestrukturyzowanych, takich jak pliki multimedialne, dokumenty czy kopie zapasowe. Dane są dostępne poprzez interfejs REST API usługi Azure Storage. Wykorzystywano je do przechowywania zdjęć profilowych użytkownika oraz multimedii (zdjęcia i filmy) ze spotów i forum.

- **Jira**

To narzędzie firmy Atlassian do zarządzania pracami nad projektem w metodach zwinnych. Do Backlogu wpisywano zadania, a na tablicy Kanbanowej rejestrowano ich statusy oraz poświęcony czas.

- **GitHub**

Zdalne repozytorium służące do przechowywania i wersjonowania kodu aplikacji. Zamieszczono tam kod naszego projektu. Do każdego zadania tworzone osobną gałąź z właściwą nazwą, a po zakończeniu prac przeprowadzano review kodu. Następnie łączono ją do głównej gałęzi deweloperskiej.

- **GitHub Actions**

To narzędzie do implementacji procesów CI/CD na platformie GitHub, które umożliwiają automatyczne testowanie lub wdrażanie kodu. Uruchamiają się w reakcji na różne operacje w repozytorium, na przykład przesłanie zmian

na wybraną gałąź. Stosowano je do automatycznego testowania i budowania projektu po każdorazowym wprowadzeniu zmian.

- **GitHub Copilot**

To narzędzie sztucznej inteligencji będące asystentem programisty. W projekcie analizuje plik oraz pliki powiązane. Wykorzystywano go podczas [review kodu](#). Copilot skanuje wszystkie pliki i w komentarzach opisuje sugerowane zmiany lub potencjalne błędy.

- **Discord**

Darmowa platforma komunikacyjna. Umożliwia udostępnienie obrazu z ekranu, komunikację głosową oraz tekstową, jak i również przesyłanie plików. Stosowano go do spotkań, na których omawiano sprawy dotyczące projektu.

- **Messenger**

Komunikator będący usługą Facebooka. Daje możliwość tworzenia czatów grupowych lub prywatnych, a także udostępniania plików. Używano go do ustalania spotkań na Discordzie oraz szybkiej komunikacji.

- **Postman**

To narzędzie służące do testowania endpointów [API](#). Pozwala grupować pytania w kolekcje, wysyłać ich różne typy oraz analizować odpowiedzi z serwera. Wykorzystywano go do testowania stworzonych endpointów oraz debugowania.

- **Figma**

Narzędzie chmurowe do projektowania interfejsów użytkownika ([UI](#)). Umożliwia zespołowe tworzenie w pełni interaktywnych prototypów. Wykonano w nim projekty ekranów naszej aplikacji.

- **Visual Paradigm**

To narzędzie do tworzenia różnych diagramów stosowanych w inżynierii oprogramowania, takich jak [UML](#)( [3]) czy [BPMN](#)( [4]). Zrobiono w nim diagram przypadków użycia.

- **Xmind**

Narzędzie służące do tworzenia mapy myśli. Wykorzystano je w celu lepszego zrozumienia problemów poprzez przeniesienie ich na diagram.

## 3.4 Zasoby i ograniczenia

### 3.4.1 Zasoby

- **Specjalizacja członków zespołu** — wszyscy członkowie zespołu projektowego specjalizują się w aplikacjach internetowych.
- **Dostęp do przedstawiciela grupy docelowej** — jeden z członków zespołu (Adam) jest [droniarzem foto/video](#).
- **Status studenta** — fakt bycia studentem zapewnia dostęp do wersji premium wielu usług (Figma Education, GitHub PRO).
- **Oprogramowanie zapewniane przez PJATK** - uczelnia zapewnia dostęp do pakietu JetBrains oraz usług firmy Microsoft (OneDrive).

### 3.4.2 Ograniczenia

- **Ograniczenia czasowe** — projekt jest ograniczony harmonogramem akademickim i terminem oddania pracy dyplomowej, co wymagało wysokiego tempa realizacji oraz sprawnej komunikacji w zespole.
- **Ograniczenia budżetowe** — projekt nie posiada finansowania i w związku z tym korzystano z rozwiązań darmowych oraz open source.

### 3.4.3 Usługi zewnętrzne

Niniejszy rozdział zawiera spis zewnętrznych [API](#) oraz usług użytych w projekcie.

KARTA USŁUGI ZEWNĘTRZNEJ	
<b>Identyfikator:</b>	UZ1
<b>Nazwa:</b>	GitHub Actions (CI) <a href="#">[5]</a>
<b>Opis:</b>	Uruchomienia pipeline'ów CI/CD dla repozytorium GitHub.
<b>Limit:</b>	3000 min/mies.

**Tabela 3.1:** Usługa zewnętrzna: GitHub Actions (CI)

KARTA USŁUGI ZEWNĘTRZNEJ	
<b>Identyfikator:</b>	UZ2
<b>Nazwa:</b>	Azure Blob Storage <a href="#">[6]</a>
<b>Opis:</b>	Magazyn plików (m.in. zdjęcia spotów, załączniki z czatu).
<b>Limit:</b>	1 GB/mies.

**Tabela 3.2:** Usługa zewnętrzna: Azure Blob Storage

KARTA USŁUGI ZEWNĘTRZNEJ	
<b>Identyfikator:</b>	UZ3
<b>Nazwa:</b>	Mailtrap <a href="#">[7]</a>
<b>Opis:</b>	Środowisko testowe SMTP oraz Email API do wysyłki maili.
<b>Limit:</b>	150 maili/dzień

**Tabela 3.3:** Usługa zewnętrzna: Mailtrap

KARTA USŁUGI ZEWNĘTRZNEJ	
<b>Identyfikator:</b>	UZ4

<b>Nazwa:</b>	LocationIQ [8]
<b>Opis:</b>	Geokodowanie adresu przy dodawaniu nowych spotów.
<b>Limit:</b>	5 000 zapytań/dzień

**Tabela 3.4:** Usługa zewnętrzna: LocationIQ

KARTA USŁUGI ZEWNĘTRZNEJ	
<b>Identyfikator:</b>	UZ5
<b>Nazwa:</b>	Google Maps (Maps URLs) [9]
<b>Opis:</b>	Otwieranie nawigacji w aplikacji Map Google (deep link/URL).
<b>Limit:</b>	Brak limitu w ramach dokumentowanego sposobu użycia.

**Tabela 3.5:** Usługa zewnętrzna: Google Maps (Maps URLs)

KARTA USŁUGI ZEWNĘTRZNEJ	
<b>Identyfikator:</b>	UZ6
<b>Nazwa:</b>	OpenFreeMap [10]
<b>Opis:</b>	Publiczny serwer kafelków do renderu mapy na froncie.
<b>Limit:</b>	30 000 zapytań/mies.

**Tabela 3.6:** Usługa zewnętrzna: OpenFreeMap

KARTA USŁUGI ZEWNĘTRZNEJ	
<b>Identyfikator:</b>	UZ7

<b>Nazwa:</b>	Open-Meteo [11]
<b>Opis:</b>	Prognozy pogody wyświetlane dla spotów.
<b>Limit:</b>	10 000 zapytań/dzień

**Tabela 3.7:** Usługa zewnętrzna: Open-Meteo

KARTA USŁUGI ZEWNĘTRZNEJ	
<b>Identyfikator:</b>	UZ8
<b>Nazwa:</b>	Tenor GIF API [12]
<b>Opis:</b>	Wyszukiwanie GIF-ów w czacie.
<b>Limit:</b>	1 zapytanie na sekundę; brak ogólnego limitu dziennego.

**Tabela 3.8:** Usługa zewnętrzna: Tenor GIF API

KARTA USŁUGI ZEWNĘTRZNEJ	
<b>Identyfikator:</b>	UZ9
<b>Nazwa:</b>	Where the ISS at? [13]
<b>Opis:</b>	HTTP API z bieżącą pozycją satelity, używane pomocniczo.
<b>Limit:</b>	1 zapytanie na sekundę; brak ogólnego limitu dziennego.

**Tabela 3.9:** Usługa zewnętrzna: Where the ISS at?

### 3.5 Analiza ryzyka

# Rozdział 4

## Analiza wymagań

Niniejszy rozdział zawiera analizę wymagań postawionych systemowi.

Do określenia priorytetów realizacji wymagań skorzystano z metody MoSCoW. Metoda MoSCoW jest techniką priorytetyzacji wymagań. Polega ona na przypisaniu każdemu wymaganiu jednej z czterech kategorii priorytetu, określających jego znaczenie dla minimalnie użytecznej wersji systemu.

W niniejszej pracy przyjęto następującą interpretację priorytetów MoSCoW:

**M – *Must have*** wymagania krytyczne dla systemu. Muszą zostać zrealizowane w bieżącej wersji, aby system mógł zostać uznany za ukończony i spełniał podstawowe cele biznesowe.

**S – *Should have*** wymagania bardzo ważne. Powinny zostać zrealizowane, jeśli pozwolą na to dostępne zasoby (czas, zespół), jednak w sytuacji konieczności ograniczenia zakresu mogą zostać przesunięte do kolejnego wydania.

**C – *Could have*** wymagania opcjonalne, „mile widziane”. Zwiększały wygodę, kompletność lub atrakcyjność systemu, ale ich brak nie uniemożliwia osiągnięcia głównych celów projektu.

**W – *Won't have this time*** wymagania świadomie odłożone. Zostały zidentyfikowane, jednak nie będą realizowane w obecnym zakresie projektu (bieżącej wersji systemu); mogą stanowić bazę dla przyszłego rozwoju rozwiązania.

W dalszej części rozdziału każdy opis wymagania zawiera przypisany priorytet MoSCoW zgodnie z powyższą klasyfikacją.

Ponadto wymagania mogą mieć jeden z dwóch statusów realizacji:

**Zrealizowano** – wymaganie zostało zrealizowane.

**Anulowano** – wymaganie zostało anulowane.

## 4.1 Przypadki użycia

### 4.1.1 Aktorzy

Niniejszy rozdział przedstawia aktorów wraz z opisami.

**Użytkownik systemu** - Reprezentuje każdą osobę korzystającą z aplikacji.

**Użytkownik niezalogowany** - Gość przeglądający publiczne treści (mapa, spoty, forum): może się zarejestrować lub zalogować.

**Użytkownik zalogowany** - Ma dostęp do wszystkich darmowych funkcjonalności aplikacji. Zarządza kontem i ulubionymi spotami, dodaje posty i komentarze, korzysta z czatu.

**Użytkownik premium** - Użytkownik z wykupioną subskrypcją: ma dostęp do funkcji premium np. oznaczenie stref **PANSA** na mapie.

**System Finansowo-księgowy** - zewnętrzny system do prowadzenia księgowości, wystawiania faktur oraz rozliczania płatności.

**Usługa SMTP** - usługa Simple Mail Transfer Protocol wykorzystywana do wysyłania wiadomości e-mail.

**Bramka Płatnicza** - usługa obsługująca płatności elektroniczne (karta płatnicza, BLIK itp.).

**Usługa OAuth** - usługa uwierzytelniania i autoryzacji użytkowników z wykorzystaniem zewnętrznych dostawców tożsamości.

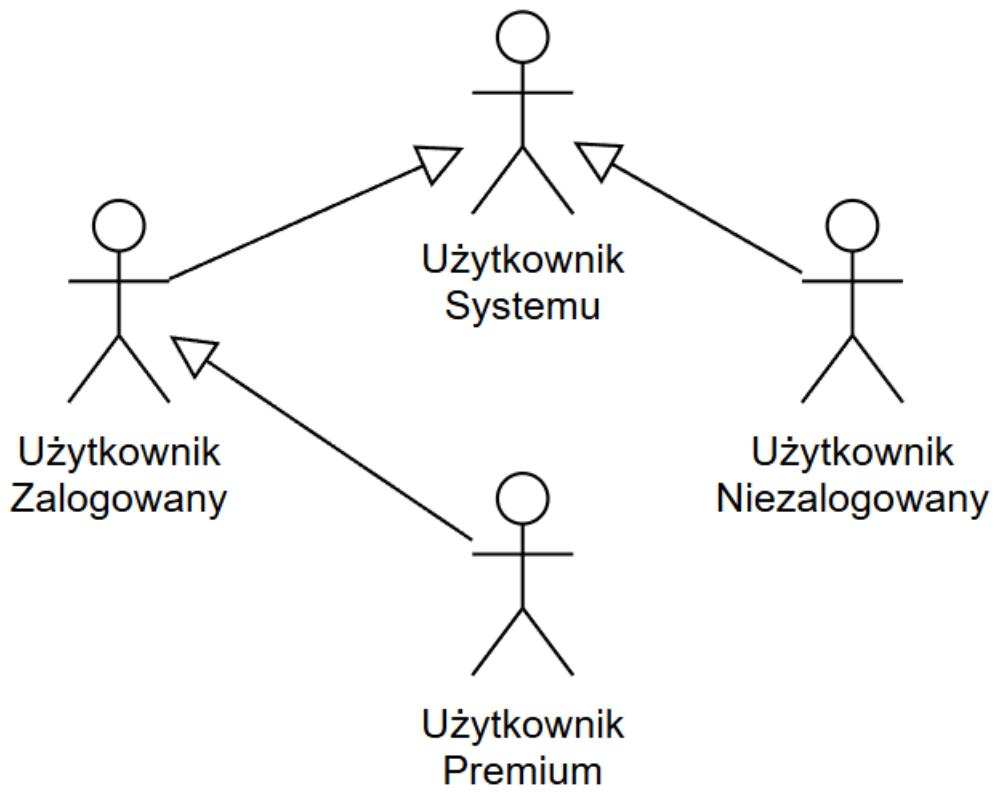
**Usługa do przechowywania plików w chmurze** - magazyn plików w chmurze służący do przechowywania załączników i multimedii użytkowników.

**Usługa do wyświetlania mapy** - zewnętrzne API dostarczające kafelki map, nawigację oraz dane geolokalizacyjne.

**Usługa danych pogodowych** - usługa udostępniająca bieżące warunki pogodowe oraz prognozy dla wybranych lokalizacji.

**Usługa do GIF'ów** - serwis umożliwiający wyszukiwanie i osadzanie animowanych obrazów GIF w aplikacji.

**Usługa do określania strefy czasowej** - usługa ustalająca strefę czasową spora podstawie jego współrzędnych geograficznych.

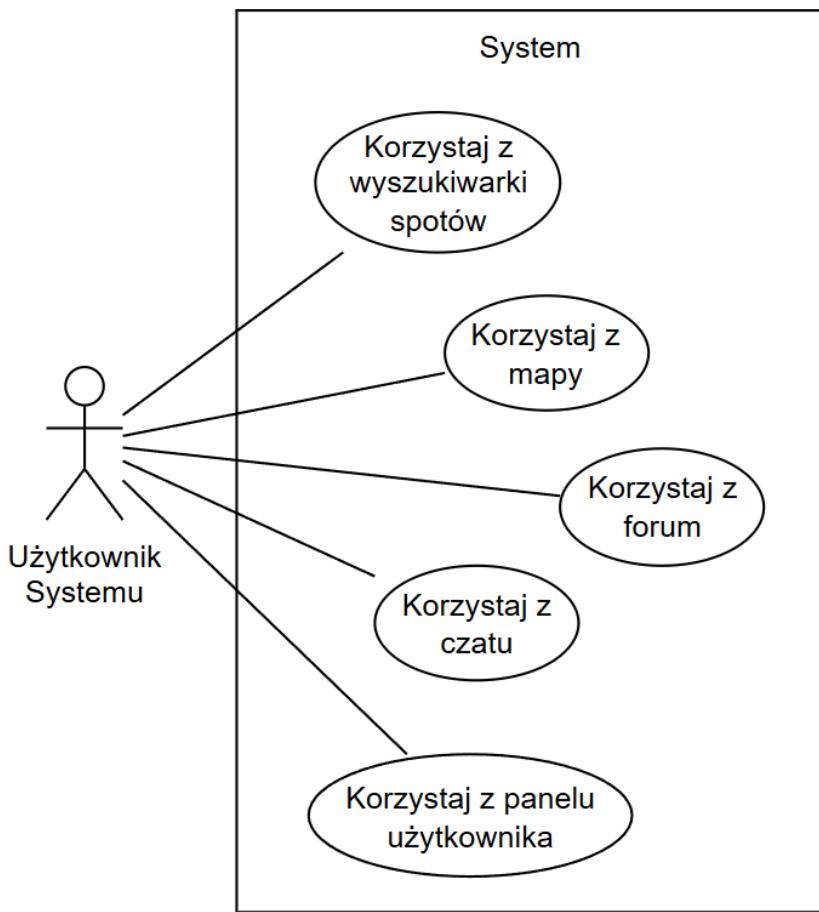


Rysunek 4.1: Diagram hierarchii użytkowników systemu

Na diagramie przedstawiono hierarchię aktorów systemu reprezentujących użytkownika. Podstawową rolą jest Użytkownik systemu, która reprezentuje każdą osobę korzystającą z aplikacji. Z niej dziedziczą dwie bardziej szczegółowe role: Użytkownik niezalogowany (ma dostęp tylko do funkcji publicznych) oraz Użytkownik zalogowany (posiada konto i dostęp do funkcji wymagających uwierzytelnienia). Użytkownik premium jest wyspecjalizowaną wersją użytkownika zalogowanego i oprócz standardowych możliwości ma także dostęp do opcji premium.

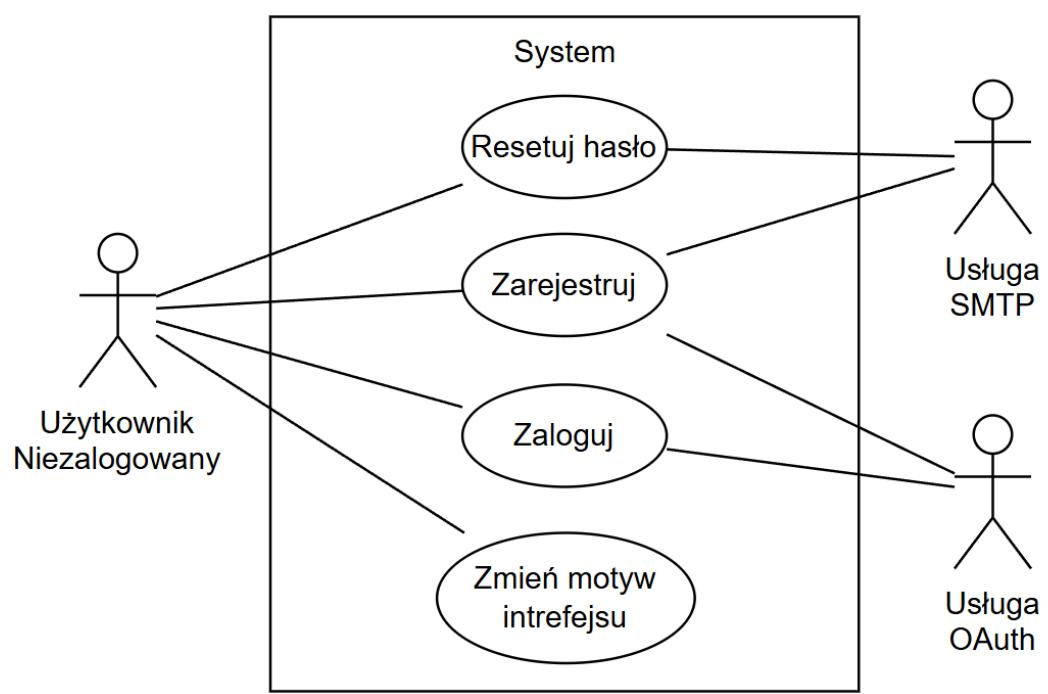
#### **4.1.2 Diagramy przypadków użycia**

Niniejszy rozdział przedstawia diagramy przypadków użycia.

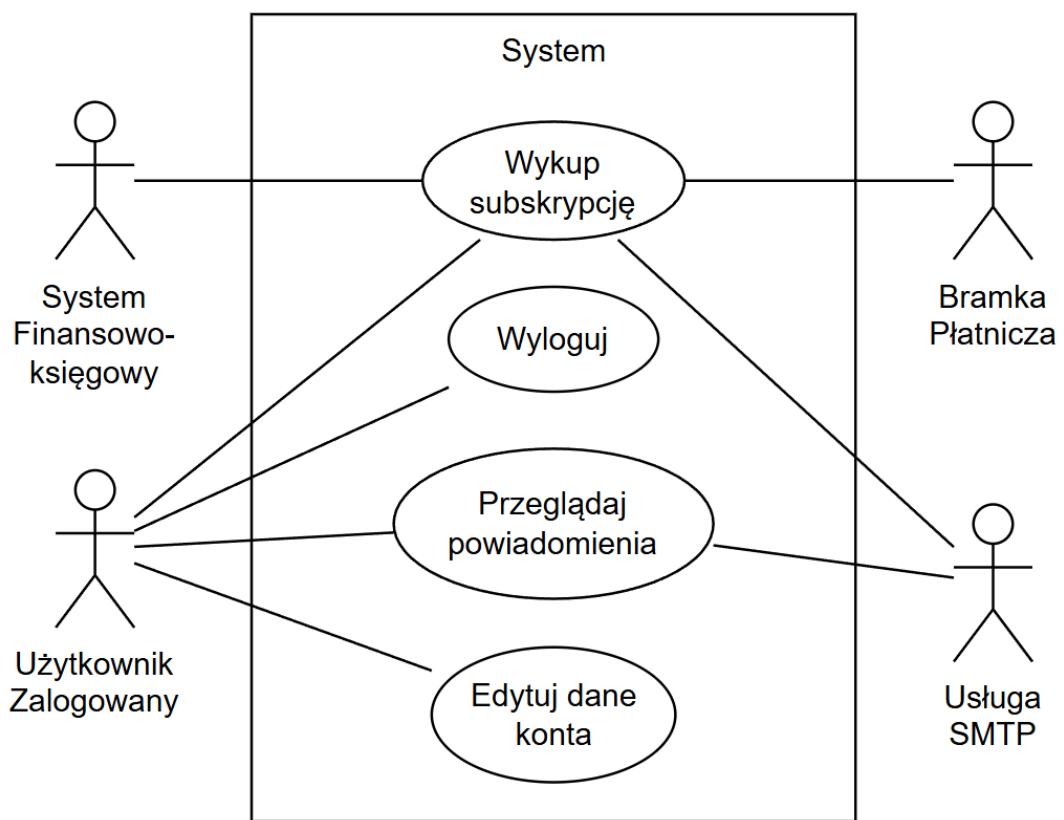


Rysunek 4.2: Wysokopoziomowy diagram przypadków użycia

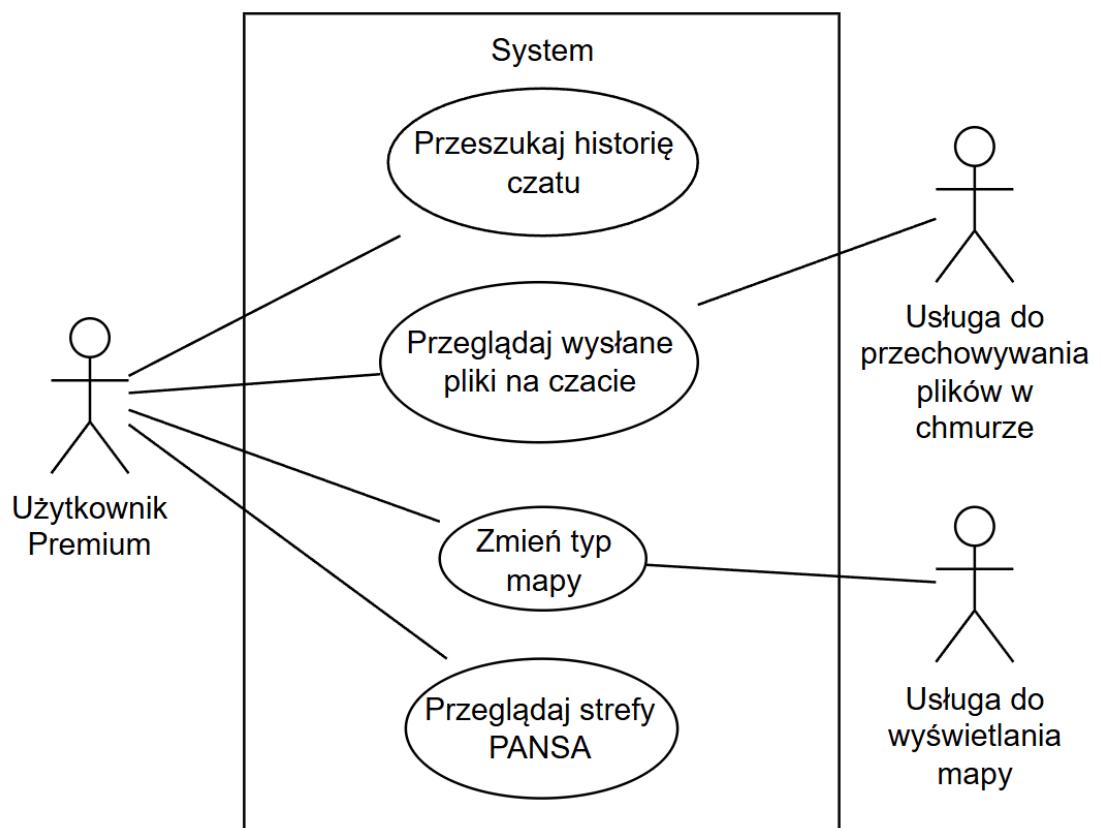
Diagram przedstawia podstawowe interakcje użytkownika z systemem. Na jego podstawie zespół projektowy podzielił architekturę aplikacji na 5 modułów: wyszukiwarkę spotów, mapę spotów, forum, czat oraz profil użytkownika. Pozostałe diagramy są bardziej szczegółowe.



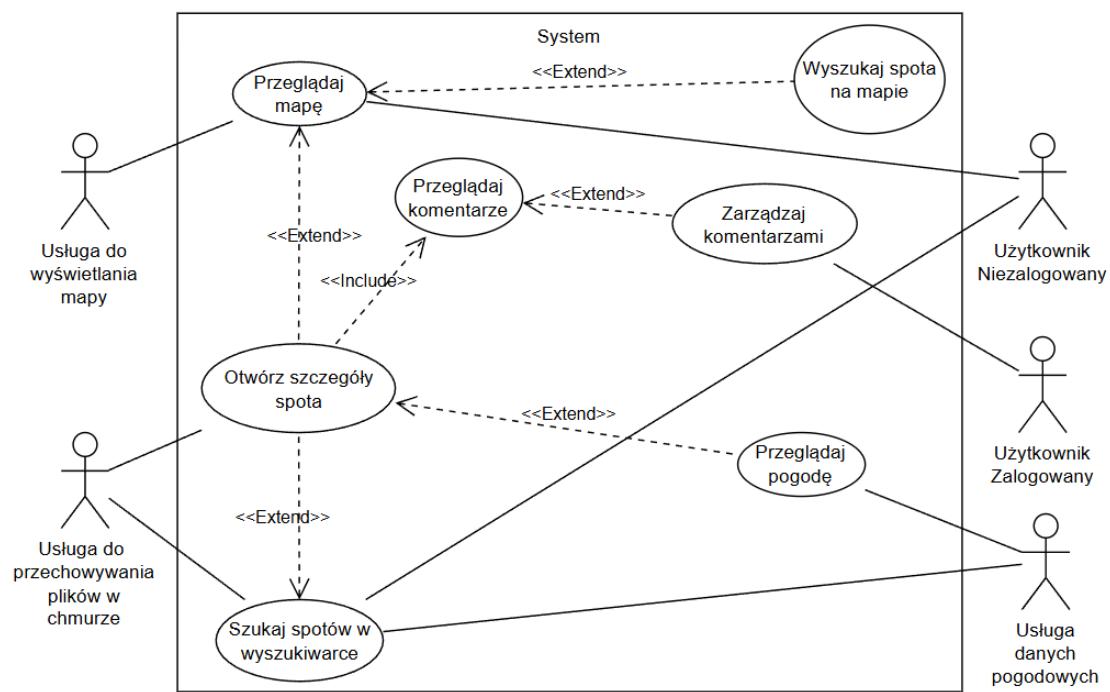
Rysunek 4.3: Diagram przypadków użycia dla użytkownika niezalogowanego



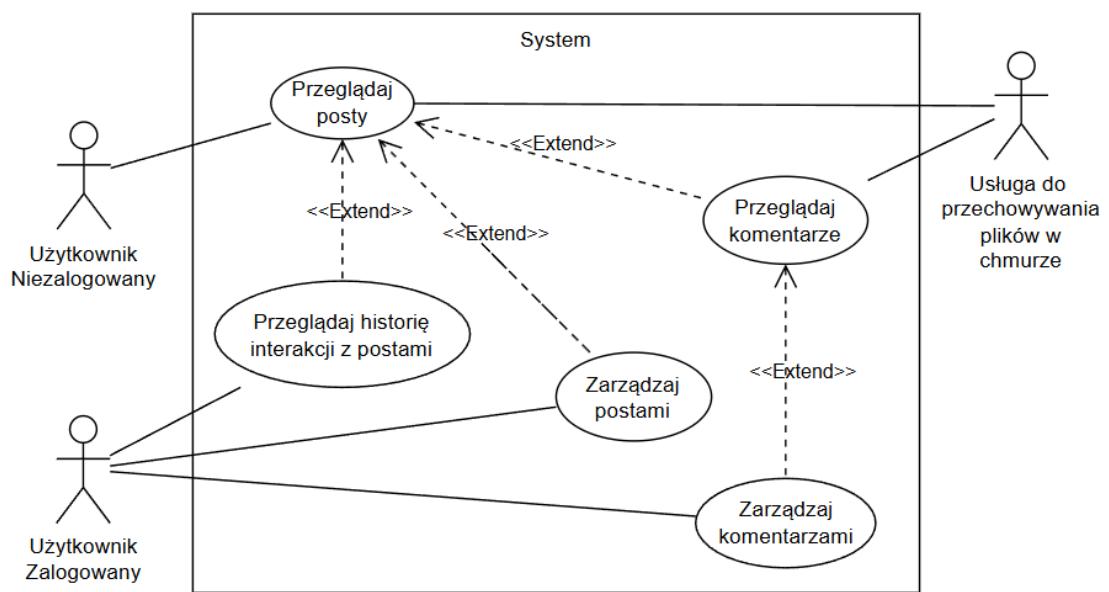
Rysunek 4.4: Diagram przypadków użycia dla użytkownika zalogowanego



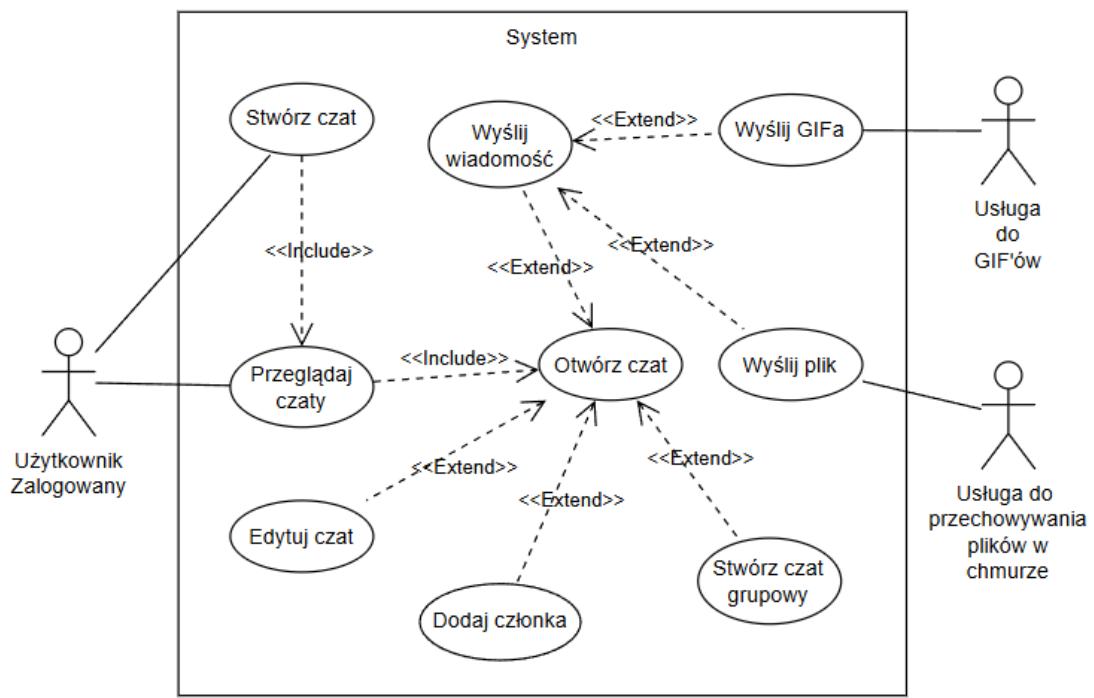
Rysunek 4.5: Diagram przypadków użycia dla użytkownika premium



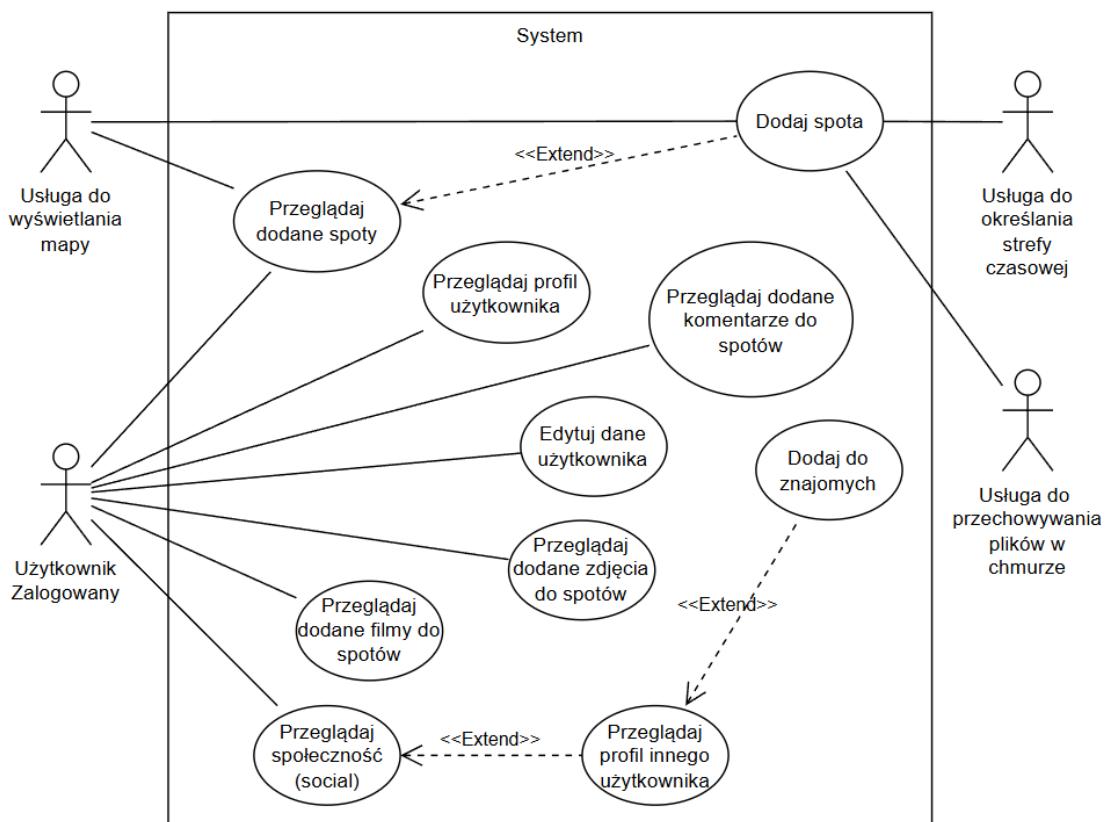
Rysunek 4.6: Diagram przypadków użycia wyszukiwarki spotów oraz mapy



Rysunek 4.7: Diagram przypadków użycia forum



Rysunek 4.8: Diagram przypadków użycia czatu



Rysunek 4.9: Diagram przypadków użycia profilu użytkownika

### 4.1.3 Scenariusze przypadków użycia

#### 4.1.3.1 Scenariusze przypadków użycia – funkcje ogólne

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU1
<b>Nazwa:</b>	Rejestracja użytkownika
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik niezalogowany
<b>Opis:</b>	Użytkownik zakłada konto poprzez formularz rejestracji.

<b>Warunki wstępne:</b>	Użytkownik znajduje się na stronie z formularzem rejestracji.
<b>Warunki końcowe:</b>	Użytkownik posiada konto w systemie.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wypełnia formularz rejestracyjny.</li> <li>2. Użytkownik naciska przycisk rejestracji.</li> <li>3. System tworzy konto użytkownika.</li> <li>4. System loguje użytkownika i przenosi go na ostatnio doowiedzoną podstronę.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>1a. Podane dane są niepoprawne – system wyświetla komunikat o błędzie oraz podświetla pola wymagające poprawy.</li> <li>2a. Nazwa użytkownika jest już zajęta – system wyświetla komunikat o błędzie.</li> <li>2b. Adres email jest już zajęty – system wyświetla komunikat o błędzie.</li> </ol>

**Tabela 4.1:** Scenariusz przypadku użycia: Rejestracja użytkownika

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU2
<b>Nazwa:</b>	Logowanie użytkownika
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik niezalogowany
<b>Opis:</b>	Użytkownik loguje się do systemu, podając login i hasło.
<b>Warunki wstępne:</b>	Użytkownik znajduje się na stronie logowania.

<b>Warunki końcowe:</b>	Użytkownik jest zalogowany i przeniesiony na ostatnio do-wiedzoną podstronę.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wypełnia formularz logowania.</li> <li>2. Użytkownik naciska przycisk logowania.</li> <li>3. System loguje użytkownika i przenosi go na ostatnio do-wiedzoną podstronę.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li value="2">2a. Podane dane są niepoprawne – system wyświetla komunikat o błędzie.</li> </ol>

**Tabela 4.2:** Scenariusz przypadku użycia: Logowanie użytkownika

<b>KARTA SCENARIUSZA PRZYPADKU UŻYCIA</b>	
<b>Identyfikator:</b>	PU3
<b>Nazwa:</b>	Wykupienie subskrypcji premium
<b>Priorytet:</b>	Niski
<b>Aktorzy:</b>	Użytkownik zalogowany, Bramka płatnicza, System finansowo-księgowy
<b>Opis:</b>	Użytkownik opłaca subskrypcję premium w celu uzyskania dodatkowych funkcji.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i znajduje się w module subskrypcji.
<b>Warunki końcowe:</b>	Subskrypcja premium jest aktywna, a użytkownik ma dostęp do funkcji premium.

<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wybiera plan subskrypcji.</li> <li>2. Użytkownik przechodzi do bramki płatniczej.</li> <li>3. Użytkownik podaje dane płatnicze i zatwierdza transakcję.</li> <li>4. Bramka płatnicza przetwarza płatność i zwraca wynik do systemu.</li> <li>5. System zapisuje informację o opłaconej subskrypcji i aktualizuje uprawnienia.</li> <li>6. System generuje wpis w systemie finansowo-księgowym.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>4a. Płatność nie powiodła się – system informuje użytkownika i umożliwia ponowną próbę.</li> <li>5a. W czasie aktualizacji subskrypcji wystąpił błąd – system cofa zmiany i wyświetla komunikat o problemie.</li> </ol>

**Tabela 4.3:** Scenariusz przypadku użycia: Wykupienie subskrypcji premium

<b>KARTA SCENARIUSZA PRZYPADKU UŻYCIA</b>	
<b>Identyfikator:</b>	PU4
<b>Nazwa:</b>	Resetowanie hasła
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik niezalogowany, Usługa SMTP
<b>Opis:</b>	Użytkownik inicjuje reset hasła, aby odzyskać dostęp do konta.
<b>Warunki wstępne:</b>	Użytkownik znajduje się na ekranie resetu hasła.
<b>Warunki końcowe:</b>	Użytkownik otrzymuje wiadomość e-mail z linkiem do ustalenia nowego hasła.

<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wpisuje adres e-mail powiązany z kontem.</li> <li>2. Użytkownik zatwierdza żądanie resetu hasła.</li> <li>3. System generuje token resetu hasła.</li> <li>4. System wysyła e-mail z linkiem do zmiany hasła.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>2a. Nie istnieje konto dla podanego adresu – system wyświetla komunikat o błędzie.</li> <li>4a. Występuje błąd połączenia z usługą SMTP – system informuje użytkownika o problemie technicznym.</li> </ol>

**Tabela 4.4:** Scenariusz przypadku użycia: Resetowanie hasła

<b>KARTA SCENARIUSZA PRZYPADKU UŻYCIA</b>	
<b>Identyfikator:</b>	PU5
<b>Nazwa:</b>	Zmiana hasła w ustawieniach konta
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik zalogowany
<b>Opis:</b>	Użytkownik zmienia hasło do konta z poziomu ustawień profilu.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i znajduje się na ekranie zmiany danych konta.
<b>Warunki końcowe:</b>	Hasło do konta użytkownika zostało zaktualizowane.

<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wpisuje aktualne hasło.</li> <li>2. Użytkownik wpisuje nowe hasło i powtarza je.</li> <li>3. Użytkownik zatwierdza formularz zmiany hasła.</li> <li>4. System zapisuje nowe hasło i informuje o powodzeniu operacji.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>3a. Aktualne hasło jest nieprawidłowe – system wyświetla komunikat i nie zapisuje zmian.</li> <li>3b. Nowe hasło nie spełnia wymagań bezpieczeństwa – system informuje o błędzie i podświetla pola do poprawy.</li> </ol>

**Tabela 4.5:** Scenariusz przypadku użycia: Zmiana hasła w ustawieniach konta

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU6
<b>Nazwa:</b>	Wylogowanie użytkownika
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik zalogowany
<b>Opis:</b>	Użytkownik wylogowuje się z aplikacji.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany.
<b>Warunki końcowe:</b>	Sesja użytkownika została zakończona, użytkownik widzi stronę główną dla niezalogowanych.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wybiera opcję wylogowania z menu.</li> <li>2. System unieważnia token dostępu użytkownika.</li> <li>3. System przenosi użytkownika na stronę główną aplikacji.</li> </ol>

<b>Alternatywne przepływy zdarzeń:</b>	Brak istotnych alternatywnych przepływów.
--	---

**Tabela 4.6:** Scenariusz przypadku użycia: Wylogowanie użytkownika

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU7
<b>Nazwa:</b>	Przeglądanie powiadomień
<b>Priorytet:</b>	Niski
<b>Aktorzy:</b>	Użytkownik zalogowany
<b>Opis:</b>	Użytkownik przegląda listę powiadomień.
<b>Warunki wstępne:</b>	Użytkownik jest na ekranie centra powiadomień.
<b>Warunki końcowe:</b>	Powiadomienia zostały wyświetcone, a wybrane oznaczone jako przeczytane.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>System wyświetla powiadomienia w odwróconym porządku chronologicznym.</li> <li>Użytkownik otwiera wybrane powiadomienie.</li> <li>System oznacza powiadomienie jako przeczytane i ewentualnie przenosi użytkownika do powiązanego widoku.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>System nie może pobrać powiadomień (błąd serwera) – użytkownik otrzymuje komunikat o błędzie i może spróbować ponownie.</li> </ol>

**Tabela 4.7:** Scenariusz przypadku użycia: Przeglądanie powiadomień

#### 4.1.3.2 Scenariusze przypadków użycia dla funkcji premium

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU8
<b>Nazwa:</b>	Przeszukiwanie historii czatu
<b>Priorytet:</b>	Niski
<b>Aktorzy:</b>	Użytkownik premium
<b>Opis:</b>	Użytkownik wyszukuje konkretne wiadomości w historii czatu.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany jako użytkownik premium i znajduje się w widoku czatu.
<b>Warunki końcowe:</b>	Wiadomości spełniające kryteria wyszukiwania zostały wyświetlane.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik otwiera pole wyszukiwania historii w czacie.</li> <li>2. Użytkownik wpisuje frazę lub filtr (np. zakres dat, autor).</li> <li>3. System filtryuje wiadomości zgodnie z kryteriami.</li> <li>4. System prezentuje listę dopasowanych fragmentów rozmowy.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<p>4a. Nie znaleziono wiadomości spełniających kryteria – system informuje o braku wyników wyszukwiania.</p>

**Tabela 4.8:** Scenariusz przypadku użycia: Przeszukiwanie historii czatu

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU9
<b>Nazwa:</b>	Przeglądanie wysłanych plików na czacie

<b>Priorytet:</b>	Niski
<b>Aktorzy:</b>	Użytkownik premium, Usługa do przechowywania plików w chmurze
<b>Opis:</b>	Użytkownik przegląda listę plików wysłanych w ramach czatów.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany jako użytkownik premium.
<b>Warunki końcowe:</b>	Użytkownik widzi listę wysłanych plików i może przechodzić do powiązanych czatów.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik otwiera sekcję „Wysłane pliki”.</li> <li>2. System pobiera metadane plików z usługi przechowywania.</li> <li>3. System wyświetla listę plików z podstawowymi informacjami (nazwa, typ, data).</li> <li>4. Użytkownik wybiera plik, aby otworzyć go lub przejść do powiązanego czatu.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<p>4a. Doszło do błędu podczas pobierania pliku - system wyświetla odpowiedni komunikat.</p>

**Tabela 4.9:** Scenariusz przypadku użycia: Przeglądanie wysłanych plików na czacie

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU10
<b>Nazwa:</b>	Zmiana typu mapy
<b>Priorytet:</b>	Niski

<b>Aktorzy:</b>	Użytkownik premium, Usługa do wyświetlania mapy
<b>Opis:</b>	Użytkownik zmienia typ mapy (np. standardowa, satelitarna, hybrydowa).
<b>Warunki wstępne:</b>	Użytkownik premium jest na ekranie mapy.
<b>Warunki końcowe:</b>	Mapa jest wyświetlana w wybranym typie.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik otwiera ustawienia widoku mapy.</li> <li>2. Użytkownik wybiera typ mapy z dostępnej listy.</li> <li>3. System przełącza widok mapy na wybrany typ.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<p>3a. Wybrany typ mapy nie jest dostępny (błąd usługi mapowej) – system przywraca poprzedni typ i informuje o błędzie.</p>

**Tabela 4.10:** Scenariusz przypadku użycia: Zmiana typu mapy

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU11
<b>Nazwa:</b>	Przeglądanie stref PANSA
<b>Priorytet:</b>	Niski
<b>Aktorzy:</b>	Użytkownik premium, Usługa do wyświetlania mapy
<b>Opis:</b>	Użytkownik wyświetla na mapie strefy przestrzeni powietrznej PANSA.
<b>Warunki wstępne:</b>	Użytkownik premium ma otwarty moduł mapy.

<b>Warunki końcowe:</b>	Strefy PANSA zostały zwizualizowane na mapie.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik włącza warstwę „Strefy PANSA”.</li> <li>2. System pobiera dane o strefach.</li> <li>3. System nakłada kontury stref na mapę.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li value="2">2a. Dane o strefach są chwilowo niedostępne – system komunikuje problem i nie włącza warstwy.</li> </ol>

**Tabela 4.11:** Scenariusz przypadku użycia: Przeglądanie stref PANSA

#### 4.1.3.3 Scenariusze przypadków użycia dla wyszukiwarki

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU12
<b>Nazwa:</b>	Wyszukiwanie spota w globalnej wyszukiwarce
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik niezalogowany, Usługa do wyświetlania mapy, Usługa do pogody
<b>Opis:</b>	Użytkownik wyszukuje spedy za pomocą globalnej wyszukiwarki w aplikacji.
<b>Warunki wstępne:</b>	Użytkownik znajduje się na stronie głównej z wyszukiwarką.
<b>Warunki końcowe:</b>	Użytkownik otrzymuje listę znalezionych spotów.

<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wpisuje frazę w globalnej wyszukiwarce.</li> <li>2. System wyszukuje spedy spełniające kryteria.</li> <li>3. System wyświetla listę wyników.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>3a. System informuje o braku wyników spełniających kryteria.</li> </ol>

**Tabela 4.12:** Scenariusz przypadku użycia: Wyszukiwanie spota w globalnej wyszukiwarce

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU13
<b>Nazwa:</b>	Przejście do spota na mapie z wyszukiwarki
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik niezalogowany
<b>Opis:</b>	Użytkownik przechodzi z wyników wyszukiwarki do widoku mapy ustawionego na konkretny spot.
<b>Warunki wstępne:</b>	Wyświetlona jest lista wyników wyszukiwania spotów.
<b>Warunki końcowe:</b>	Mapa jest przybliżona do wybranego spota, a jego szczegóły są dostępne.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wybiera spota z listy wyników.</li> <li>2. System przełącza widok na moduł mapy.</li> <li>3. System ustawia mapę na lokalizację spota i otwiera jego szczegóły.</li> </ol>

<b>Alternatywne przepływy zdarzeń:</b>	Brak istotnych alternatywnych przepływów.
--	---

**Tabela 4.13:** Scenariusz przypadku użycia: Przejście do spota na mapie z wyszukiwarki

#### 4.1.3.4 Scenariusze przypadków użycia dla mapy

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU14
<b>Nazwa:</b>	Przeglądanie mapy spotów
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik niezalogowany, Usługa do wyświetlania mapy
<b>Opis:</b>	Użytkownik przegląda mapę spotów.
<b>Warunki wstępne:</b>	Użytkownik znajduje się w module mapy.
<b>Warunki końcowe:</b>	Mapa ze spotami została wyświetlona, a użytkownik może przybliżać, oddalać i przesuwać widok.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>System inicjuje widok mapy z domyślnym obszarem.</li> <li>System pobiera listę spotów.</li> <li>System rysuje znaczniki spotów na mapie.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>a. Usługa mapy jest niedostępna – system wyświetla komunikat o błędzie.</li> </ol>

**Tabela 4.14:** Scenariusz przypadku użycia: Przeglądanie mapy spotów

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU15
<b>Nazwa:</b>	Otwarcie szczegółów spota
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik niezalogowany, Użytkownik zalogowany
<b>Opis:</b>	Użytkownik otwiera widok szczegółów wybranego spota.
<b>Warunki wstępne:</b>	Użytkownik widzi mapę spotów.
<b>Warunki końcowe:</b>	Wyświetlony został widok szczegółów spota z podstawowymi informacjami oraz jego lokalizacją na mapie.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wybiera spota z mapy.</li> <li>2. System pobiera dane szczegółowe spota (informacje opisowe, lokalizacja).</li> <li>3. System otwiera widok szczegółów spota.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>2a. Spot nie istnieje (został usunięty lub ukryty) – system informuje użytkownika i powraca do poprzedniego widoku.</li> <li>2b. Wystąpił błąd podczas pobierania danych spota – system wyświetla komunikat o błędzie i umożliwia ponowną próbę.</li> </ol>

**Tabela 4.15:** Scenariusz przypadku użycia: Otwarcie szczegółów spota

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU16
<b>Nazwa:</b>	Przeglądanie komentarzy do spota
<b>Priorytet:</b>	Średni
<b>Aktorzy:</b>	Użytkownik niezalogowany

<b>Opis:</b>	Użytkownik czyta komentarze pod wybranym spotem.
<b>Warunki wstępne:</b>	Wyświetlany jest widok szczegółów spota.
<b>Warunki końcowe:</b>	Lista komentarzy do spota została wyświetlona.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. System pobiera komentarze powiązane ze spotem.</li> <li>2. System wyświetla komentarze w kolejności chronologicznej lub według popularności.</li> <li>3. Użytkownik przewija listę komentarzy.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	1a. Spot nie ma jeszcze komentarzy – system wyświetla odpowiednią informację.

**Tabela 4.16:** Scenariusz przypadku użycia: Przeglądanie komentarzy do spota

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU17
<b>Nazwa:</b>	Przeglądanie pogody na spocie
<b>Priorytet:</b>	Średni
<b>Aktorzy:</b>	Użytkownik niezalogowany, Usługa danych pogodowych
<b>Opis:</b>	Użytkownik sprawdza prognozę pogody dla lokalizacji spota.
<b>Warunki wstępne:</b>	Wyświetlany jest widok szczegółów spota.
<b>Warunki końcowe:</b>	Prognoza pogody dla spota została wyświetlona.

<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik otwiera zakładkę pogody.</li> <li>2. System wysyła zapytanie do usługi pogodowej z lokalizacją spota.</li> <li>3. System odbiera prognozę i prezentuje ją (temperatura, prędkość wiatru, opady).</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<p>2a. Usługa pogodowa jest niedostępna – system wyświetla komunikat o braku danych pogodowych.</p>

**Tabela 4.17:** Scenariusz przypadku użycia: Przeglądanie pogody na spocie

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU18
<b>Nazwa:</b>	Wyszukiwanie spota na mapie
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik niezalogowany
<b>Opis:</b>	Użytkownik wyszukuje spota po nazwie korzystając z pola wyszukiwania na mapie.
<b>Warunki wstępne:</b>	Użytkownik widzi mapę spotów.
<b>Warunki końcowe:</b>	Mapa zostaje ustawiona na wybranego spota lub listę dopasowań.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wpisuje frazę w polu wyszukiwania na mapie.</li> <li>2. System podpowiada listę pasujących spotów.</li> <li>3. Użytkownik wybiera spota z listy.</li> <li>4. System przenosi użytkownika na mapie do wybranego spota.</li> </ol>

<b>Alternatywne przepływy zdarzeń:</b>	2a. Brak wyników dla podanej frazy – system informuje użytkownika o braku dopasowań.
--	--

**Tabela 4.18:** Scenariusz przypadku użycia: Wyszukiwanie spota na mapie

#### 4.1.3.5 Scenariusze przypadków użycia dla czatu

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU19
<b>Nazwa:</b>	Utworzenie prywatnego czatu
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik zalogowany
<b>Opis:</b>	Użytkownik tworzy prywatną konwersację z innym użytkownikiem.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i znajduje się w zakładce społeczność.
<b>Warunki końcowe:</b>	Nowy czat prywatny został utworzony i wyświetlony użytkownikowi.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>Użytkownik wybiera opcję utworzenia nowego czatu.</li> <li>System tworzy nowy czat (jeśli nie istnieje).</li> <li>System otwiera widok nowego czatu.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>a. Taki czat już istnieje – system zamiast tworzyć nowy, otwiera istniejącą konwersację.</li> </ol>

**Tabela 4.19:** Scenariusz przypadku użycia: Utworzenie prywatnego czatu

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU20
<b>Nazwa:</b>	Otworzenie czatu
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik zalogowany
<b>Opis:</b>	Użytkownik otwiera wybrany czat, aby wyświetlić historię rozmowy i móc wysyłać kolejne wiadomości.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i widzi listę swoich czatów lub otrzymał powiadomienie prowadzące do czatu.
<b>Warunki końcowe:</b>	Wybrany czat został otworzony, a historia rozmowy jest wiadoma dla użytkownika.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>Użytkownik wybiera czat z listy czatów lub z powiadomienia.</li> <li>System pobiera dane czatu (uczestników, ostatnie wiadomości).</li> <li>System oznacza nieprzeczytane wiadomości na czacie jako przeczytane.</li> <li>System wyświetla widok czatu wraz z historią rozmowy.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>Czat nie jest już dostępny (np. został usunięty lub użytkownik utracił do niego dostęp) – system wyświetla komunikat o braku dostępu i powraca do listy czatów.</li> <li>Wystąpił błąd podczas pobierania danych czatu – system wyświetla komunikat o błędzie i umożliwia ponowną próbę.</li> </ol>

**Tabela 4.20:** Scenariusz przypadku użycia: Otworzenie czatu

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU21
<b>Nazwa:</b>	Utworzenie czatu grupowego
<b>Priorytet:</b>	Średni
<b>Aktorzy:</b>	Użytkownik zalogowany
<b>Opis:</b>	Użytkownik tworzy nowy czat grupowy z kilkoma uczestnikami.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i znajduje się na dowolnym czasie prywatnym.
<b>Warunki końcowe:</b>	Czat grupowy został utworzony i wyświetlony na ekranie.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wybiera opcję utworzenia czatu grupowego.</li> <li>2. Użytkownik wybiera uczestników grupy.</li> <li>3. Użytkownik zatwierdza utworzenie czatu.</li> <li>4. System tworzy czat grupowy i dodaje do niego wskazanych użytkowników.</li> <li>5. System otwiera widok nowego czatu grupowego.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<p>3a. System nie może utworzyć czatu – aplikacja informuje o błędzie.</p>

**Tabela 4.21:** Scenariusz przypadku użycia: Utworzenie czatu grupowego

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU22
<b>Nazwa:</b>	Przeglądanie listy czatów
<b>Priorytet:</b>	Wysoki

<b>Aktorzy:</b>	Użytkownik zalogowany
<b>Opis:</b>	Użytkownik przegląda listę swoich czatów prywatnych i grupowych.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i otwiera moduł czatu.
<b>Warunki końcowe:</b>	Lista czatów użytkownika została wyświetlona.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. System pobiera listę czatów użytkownika.</li> <li>2. System wyświetla listę czatów z podstawowymi informacjami.</li> <li>3. Użytkownik wybiera czat z listy.</li> <li>4. System otwiera widok wybranego czatu.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	Brak istotnych alternatywnych przepływów.

**Tabela 4.22:** Scenariusz przypadku użycia: Przeglądanie listy czatów

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU23
<b>Nazwa:</b>	Wysyłanie wiadomości na czacie
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik zalogowany
<b>Opis:</b>	Użytkownik wysyła wiadomość tekstową na czacie.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i znajduje się w widoku konkretnego czatu.

<b>Warunki końcowe:</b>	Nowa wiadomość jest zapisana i widoczna w historii czatu.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wpisuje treść wiadomości.</li> <li>2. Użytkownik wysyła wiadomość.</li> <li>3. System zapisuje wiadomość i dostarcza ją do uczestników czatu.</li> <li>4. System wyświetla wiadomość na liście wiadomości.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<p>2a. Treść wiadomości jest pusta – system blokuje wysłanie i pozostaje w tym samym widoku.</p>

**Tabela 4.23:** Scenariusz przypadku użycia: Wysyłanie wiadomości na czacie

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU24
<b>Nazwa:</b>	Wysyłanie GIF-a na czacie
<b>Priorytet:</b>	Średni
<b>Aktorzy:</b>	Użytkownik zalogowany, Usługa GIF-ów
<b>Opis:</b>	Użytkownik wysyła animację GIF w konwersacji czatowej.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i znajduje się w widoku czatu.
<b>Warunki końcowe:</b>	Wybrany GIF został dodany jako wiadomość w czacie.

<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wybiera opcję dodania GIF-a.</li> <li>2. System otwiera okno wyszukiwarki GIF-ów.</li> <li>3. Użytkownik wybiera lub wyszukuje GIF-a.</li> <li>4. Użytkownik zatwierdza wysłanie GIF-a.</li> <li>5. System dodaje GIF-a jako wiadomość na czacie.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<p>2a. Usługa GIF-ów jest niedostępna – system informuje o braku możliwości wysłania GIF-a.</p>

**Tabela 4.24:** Scenariusz przypadku użycia: Wysyłanie GIF-a na czacie

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU25
<b>Nazwa:</b>	Wysyłanie pliku na czacie
<b>Priorytet:</b>	Średni
<b>Aktorzy:</b>	Użytkownik zalogowany, Usługa do przechowywania plików w chmurze
<b>Opis:</b>	Użytkownik wysyła plik (np. zdjęcie, film) na czacie.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i znajduje się w widoku czatu.
<b>Warunki końcowe:</b>	Plik został zapisany w chmurze i powiązany z wiadomością na czacie.

<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wybiera opcję dodania pliku.</li> <li>2. Użytkownik wybiera plik z urządzenia.</li> <li>3. System przesyła plik do usługi przechowywania w chmurze.</li> <li>4. System tworzy wiadomość z odnośnikiem do pliku.</li> <li>5. System wyświetla wiadomość na liście czatu.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<p>3a. Przesyłanie pliku nie powiodło się – system informuje użytkownika i umożliwia ponowną próbę.</p>

**Tabela 4.25:** Scenariusz przypadku użycia: Wysyłanie pliku na czacie

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU26
<b>Nazwa:</b>	Edycja ustawień czatu
<b>Priorytet:</b>	Niski
<b>Aktorzy:</b>	Użytkownik zalogowany
<b>Opis:</b>	Użytkownik modyfikuje ustawienia czatu (np. nazwę, avatar, tryb powiadomień).
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i ma uprawnienia do edycji danego czatu.
<b>Warunki końcowe:</b>	Zaktualizowane ustawienia czatu są zapisane i widoczne dla uczestników.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik otwiera panel ustawień czatu.</li> <li>2. Użytkownik wprowadza zmiany (np. nazwę, opis, avatar).</li> <li>3. Użytkownik zapisuje zmiany.</li> <li>4. System waliduje dane i aktualizuje konfigurację czatu.</li> </ol>

<b>Alternatywne przepływy zdarzeń:</b>	Brak istotnych alternatywnych przepływów poza walidacja pól.
--	--

**Tabela 4.26:** Scenariusz przypadku użycia: Edycja ustawień czatu

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU27
<b>Nazwa:</b>	Dodanie członka do czatu grupowego
<b>Priorytet:</b>	Średni
<b>Aktorzy:</b>	Użytkownik zalogowany
<b>Opis:</b>	Użytkownik dodaje nowego uczestnika do czatu grupowego.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i znajduje się w czacie grupowym.
<b>Warunki końcowe:</b>	Nowy uczestnik został dodany do czatu grupowego.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>Użytkownik otwiera listę uczestników czatu grupowego.</li> <li>Użytkownik wybiera opcję dodania nowego członka.</li> <li>Użytkownik wskazuje użytkownika do dodania i zatwierdza wybór.</li> <li>System dodaje wskazanego użytkownika do czatu grupowego.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>Operacja nie powiodła się – system informuje o błędzie.</li> </ol>

**Tabela 4.27:** Scenariusz przypadku użycia: Dodanie członka do czatu grupowego

#### 4.1.3.6 Scenariusze przypadków użycia dla forum

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU28
<b>Nazwa:</b>	Przeglądanie postów na forum
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik niezalogowany, Użytkownik zalogowany
<b>Opis:</b>	Użytkownik przegląda listę postów na forum z możliwością sortowania wyników.
<b>Warunki wstępne:</b>	Użytkownik znajduje się w module forum.
<b>Warunki końcowe:</b>	Lista postów forum jest wyświetlona, a użytkownik może przechodzić do szczegółów wybranego posta.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>Użytkownik przechodzi do widoku listy postów forum.</li> <li>System pobiera listę postów i domyślnie wyświetla je w kolejności od najnowszych.</li> <li>Użytkownik wybiera sposób sortowania listy.</li> <li>System aktualizuje listę postów zgodnie z wybranym kryterium sortowania.</li> <li>Użytkownik wybiera post, który chce przeczytać.</li> <li>System otwiera szczegółowy widok wybranego posta.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>System nie może pobrać szczegółów posta – system wyświetla komunikat o błędzie.</li> </ol>

**Tabela 4.28:** Scenariusz przypadku użycia: Przeglądanie postów na forum

KARTA SCENARIUSZA PRZYPADKU UŻYCIA

<b>Identyfikator:</b>	PU29
<b>Nazwa:</b>	Wyszukiwanie postów na forum
<b>Priorytet:</b>	Średni
<b>Aktorzy:</b>	Użytkownik niezalogowany, Użytkownik zalogowany
<b>Opis:</b>	Użytkownik wyszukuje posty na forum na podstawie tytułu, kategorii, tagów oraz autora.
<b>Warunki wstępne:</b>	Użytkownik znajduje się w module forum lub na stronie wyszukiwarki postów.
<b>Warunki końcowe:</b>	Lista postów spełniających zadane kryteria wyszukiwania jest wyświetlona.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik otwiera panel wyszukiwania postów na forum.</li> <li>2. Użytkownik określa kryteria wyszukiwania (np. tytuł, kategoria, tagi, autor).</li> <li>3. Użytkownik uruchamia wyszukiwanie.</li> <li>4. System filtryuje posty zgodnie z podanymi kryteriami i wyświetla listę wyników.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<p>4a. Brak postów spełniających zadane kryteria – system wyświetla informację o braku wyników.</p> <p>4b. Wystąpił błąd podczas wyszukiwania – system wyświetla komunikat o błędzie i umożliwia ponowną próbę.</p>

**Tabela 4.29:** Scenariusz przypadku użycia: Wyszukiwanie postów na forum

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU30
<b>Nazwa:</b>	Dodanie posta na forum

<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik zalogowany, Usługa do przechowywania plików w chmurze
<b>Opis:</b>	Użytkownik publikuje nowy post na forum, określając jego treść, kategorię, tagi oraz opcjonalne załączniki.
<b>Warunki wstępne:</b>	Użytkownik znajduje się w module forum.
<b>Warunki końcowe:</b>	Nowy post jest poprawnie zapisany i widoczny na forum.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wybiera opcję dodania nowego posta.</li> <li>2. Użytkownik wpisuje tytuł i treść posta.</li> <li>3. Użytkownik wybiera kategorię posta.</li> <li>4. (Opcjonalnie) Użytkownik wybiera tagi przypisane do posta.</li> <li>5. (Opcjonalnie) Użytkownik dodaje załączniki (zdjęcia / filmy) do posta.</li> <li>6. Użytkownik publikuje posta.</li> <li>7. System zapisuje posta (oraz poprawne załączniki w chmurze) i wyświetla go na liście postów.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<p>6a. Załącznik nie może zostać zapisany lub nie spełnia wymagań (np. zbyt duży rozmiar, nieobsługiwany format) – system informuje użytkownika o błędzie, blokuje publikację posta i wymaga usunięcia lub podmiany problematycznego pliku.</p> <p>6b. Formularz zawiera błędne lub niekompletne dane (np. brak tytułu lub treści) – system wyświetla komunikat i prosi o poprawę danych przed publikacją.</p>

**Tabela 4.30:** Scenariusz przypadku użycia: Dodanie posta na forum

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU31
<b>Nazwa:</b>	Dodanie komentarza na forum
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik zalogowany
<b>Opis:</b>	Użytkownik dodaje komentarz pod postem na forum, opcjonalnie z załącznikami (zdjęcia/filmy).
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i widzi szczegóły posta.
<b>Warunki końcowe:</b>	Nowy komentarz został zapisany i jest widoczny pod postem.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>Użytkownik wpisuje treść komentarza w formularzu pod postem.</li> <li>(Opcjonalnie) Użytkownik dodaje załączniki (zdjęcia/filmy) do komentarza.</li> <li>Użytkownik publikuje komentarz.</li> <li>System zapisuje komentarz (oraz poprawne załączniki) i odświeża listę komentarzy.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<p>3a. Treść komentarza lub załączniki są niepoprawne (np. naruszają walidację) – system wyświetla komunikat o błędzie i blokuje publikację do czasu poprawy danych.</p>

**Tabela 4.31:** Scenariusz przypadku użycia: Dodanie komentarza na forum

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU32
<b>Nazwa:</b>	Przeglądanie historii interakcji z postami

<b>Priorytet:</b>	Średni
<b>Aktorzy:</b>	Użytkownik zalogowany
<b>Opis:</b>	Użytkownik przegląda historię swoich aktywności na forum (dodane posty, komentarze, reakcje).
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany.
<b>Warunki końcowe:</b>	Lista interakcji użytkownika z postami jest wyświetlona.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>Użytkownik przechodzi do sekcji historii aktywności.</li> <li>System pobiera historię interakcji użytkownika.</li> <li>System wyświetla listę interakcji z możliwością filtrowania.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	Brak istotnych alternatywnych przepływów.

**Tabela 4.32:** Scenariusz przypadku użycia: Przeglądanie historii interakcji z postami

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU33
<b>Nazwa:</b>	Zarządzanie komentarzami na forum
<b>Priorytet:</b>	Niski
<b>Aktorzy:</b>	Użytkownik zalogowany
<b>Opis:</b>	Użytkownik zarządza komentarzami pod postami forum (edytacja, usuwanie, zgłaszanie komentarzy innych użytkowników).

<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i ma dostęp do danego wątku forum.
<b>Warunki końcowe:</b>	Komentarze zostały zaktualizowane zgodnie z działaniami użytkownika.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik otwiera widok komentarzy pod postem.</li> <li>2. Użytkownik wybiera komentarz i odpowiednią akcję (edytacja, usunięcie, zgłoszenie).</li> <li>3. System weryfikuje uprawnienia użytkownika oraz zgodność akcji z jego rolą.</li> <li>4. System wykonuje wybraną akcję (np. zapisuje zmiany, usuwa komentarz lub przygotowuje zgłoszenie) i aktualizuje widok.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>2a. Użytkownik próbuje zgłosić własny komentarz – system blokuje operację i informuje, że nie można zgłaszać własnych treści.</li> <li>3a. Użytkownik nie ma wymaganych uprawnień do wykonania wybranej akcji – system blokuje operację i informuje o braku uprawnień.</li> </ol>

**Tabela 4.33:** Scenariusz przypadku użycia: Zarządzanie komentarzami na forum

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU34
<b>Nazwa:</b>	Zgłoszenie komentarza naruszającego regulamin
<b>Priorytet:</b>	Średni
<b>Aktorzy:</b>	Użytkownik zalogowany

<b>Opis:</b>	Użytkownik zgłasza komentarz na forum jako naruszający regulamin.
<b>Warunki wstępne:</b>	Użytkownik widzi komentarz w aplikacji.
<b>Warunki końcowe:</b>	Zgłoszenie komentarza zostało zapisane i trafiło do kolejki moderacyjnej.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wybiera opcję „Zgłoś komentarz”.</li> <li>2. Użytkownik wybiera kategorię naruszenia, podaje szczegóły i potwierdza zgłoszenie.</li> <li>3. System zapisuje zgłoszenie i wiąże je z komentarzem oraz zgłaszającym użytkownikiem.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>1a. Użytkownik próbuje zgłosić własny komentarz – system blokuje operację i wyświetla komunikat, że nie można zgłaszać własnych treści.</li> <li>3a. Komentarz został już wcześniej zgłoszony – system informuje użytkownika, że komentarz znajduje się już w kolejce moderacyjnej i nie zapisuje kolejnego zgłoszenia.</li> </ol>

**Tabela 4.34:** Scenariusz przypadku użycia: Zgłoszenie komentarza naruszającego regulamin

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU35
<b>Nazwa:</b>	Zgłoszenie posta na forum
<b>Priorytet:</b>	Średni
<b>Aktorzy:</b>	Użytkownik zalogowany

<b>Opis:</b>	Użytkownik zgłasza post forum jako naruszający regulamin lub tematykę.
<b>Warunki wstępne:</b>	Wyświetlony jest widok posta na forum.
<b>Warunki końcowe:</b>	Zgłoszenie posta zostało zapisane
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wybiera opcję „Zgłoś post”.</li> <li>2. Użytkownik wybiera kategorię naruszenia, podaje szczegóły i potwierdza zgłoszenie.</li> <li>3. System zapisuje zgłoszenie i oznacza post jako zgłoszony.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>1a. Użytkownik próbuje zgłosić własny post – system blokuje operację i wyświetla komunikat, że nie można zgłaszać własnych treści.</li> <li>3a. Post został już wcześniej zgłoszony – system informuje użytkownika, że post jest już zgłoszony i nie zapisuje kolejnego zgłoszenia.</li> </ol>

**Tabela 4.35:** Scenariusz przypadku użycia: Zgłoszenie posta na forum

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU36
<b>Nazwa:</b>	Przeglądanie komentarzy pod postem
<b>Priorytet:</b>	Średni
<b>Aktorzy:</b>	Użytkownik niezalogowany, Użytkownik zalogowany
<b>Opis:</b>	Użytkownik przegląda komentarze dodane pod wybranym postem na forum z możliwością zmiany kolejności ich wyświetlania.

<b>Warunki wstępne:</b>	Wyświetlany jest szczegółowy widok posta na forum.
<b>Warunki końcowe:</b>	Lista komentarzy powiązanych z postem została wyświetlona zgodnie z wybranym kryterium sortowania.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. System pobiera komentarze powiązane z wybranym postem i domyślnie wyświetla je w kolejności od najnowszych.</li> <li>2. Użytkownik wybiera sposób sortowania komentarzy.</li> <li>3. System aktualizuje listę komentarzy zgodnie z wybranym kryterium sortowania.</li> <li>4. Użytkownik przewija listę komentarzy i zapoznaje się z ich treścią.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>1a. Post nie ma jeszcze komentarzy – system wyświetla informację o braku komentarzy.</li> <li>1b. Wystąpił błąd podczas pobierania komentarzy – system wyświetla komunikat o błędzie i umożliwia ponowną próbę.</li> </ol>

**Tabela 4.36:** Scenariusz przypadku użycia: Przeglądanie komentarzy pod postem

#### 4.1.3.7 Scenariusze przypadków użycia dla panelu użytkownika

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU37
<b>Nazwa:</b>	Dodanie spota w panelu użytkownika
<b>Priorytet:</b>	Wysoki

<b>Aktorzy:</b>	Użytkownik zalogowany, Usługa do wyświetlania mapy, Usługa do przechowywania plików w chmurze
<b>Opis:</b>	Użytkownik dodaje nowy spot poprzez panel.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i znajduje się w widoku panelu użytkownika.
<b>Warunki końcowe:</b>	Nowy spot został zapisany i jest widoczny na mapie oraz w panelu użytkownika.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wybiera opcję „Dodaj spota”.</li> <li>2. Użytkownik uzupełnia podstawowe informacje o specie (nazwa, opis, tagi).</li> <li>3. Użytkownik wskazuje lokalizację spota na mapie.</li> <li>4. Użytkownik dodaje zdjęcia/filmy do spota.</li> <li>5. Użytkownik zapisuje spota.</li> <li>6. System zapisuje dane spota (oraz pliki w chmurze) i aktualizuje mapę.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	3a. Podane dane wejściowe są niepoprawne – system wyświetla komunikat i zaznacza wymagające poprawy pola.

**Tabela 4.37:** Scenariusz przypadku użycia: Dodanie spota w panelu użytkownika

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU38
<b>Nazwa:</b>	Przeglądanie profilu użytkownika
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik zalogowany

<b>Opis:</b>	Użytkownik przegląda swój profil (lista spotów, media, podstawowe dane).
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany.
<b>Warunki końcowe:</b>	Wyświetlony jest widok profilu użytkownika wraz z jego zawartością.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik otwiera swój profil.</li> <li>2. System pobiera dane profilu (informacje podstawowe, spoty, media).</li> <li>3. System wyświetla dane w odpowiednich sekcjach (spoty, zdjęcia, filmy, komentarze).</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<p>2a. Wystąpił błąd podczas pobierania danych użytkownika – system wyświetla informację o błędzie.</p>

**Tabela 4.38:** Scenariusz przypadku użycia: Przeglądanie profilu użytkownika

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU39
<b>Nazwa:</b>	Przeglądanie profilu innego użytkownika
<b>Priorytet:</b>	Średni
<b>Aktorzy:</b>	Użytkownik niezalogowany
<b>Opis:</b>	Użytkownik ogląda profil innego użytkownika (np. z mapy, forum lub społeczności).
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i ma dostęp do odnośnika do profilu innego użytkownika.

<b>Warunki końcowe:</b>	Profil innego użytkownika został wyświetlony.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wybiera odnośnik do profilu innego użytkownika.</li> <li>2. System pobiera dane profilu docelowego użytkownika.</li> <li>3. System wyświetla profil (media, podstawowe informacje).</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li value="2">2a. Wystąpił błąd podczas pobierania danych użytkownika – system wyświetla informację o błędzie.</li> </ol>

**Tabela 4.39:** Scenariusz przypadku użycia: Przeglądanie profilu innego użytkownika

<b>KARTA SCENARIUSZA PRZYPADKU UŻYCIA</b>	
<b>Identyfikator:</b>	PU40
<b>Nazwa:</b>	Dodanie użytkownika do znajomych
<b>Priorytet:</b>	Średni
<b>Aktorzy:</b>	Użytkownik zalogowany
<b>Opis:</b>	Użytkownik wysyła lub akceptuje zaproszenie do znajomych.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i przegląda profil innego użytkownika.
<b>Warunki końcowe:</b>	Relacja „znajomy” została utworzona lub zaproszenie czeka na akceptację.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik kliką przycisk „Dodaj do znajomych”.</li> <li>2. System sprawdza, czy relacja już istnieje.</li> <li>3. System tworzy nowe zaproszenie.</li> <li>4. System informuje o statusie o wysłaniu zaproszenia.</li> </ol>

<b>Alternatywne przepływy zdarzeń:</b>	Brak istotnych alternatywnych przepływów.
--	---

**Tabela 4.40:** Scenariusz przypadku użycia: Dodanie użytkownika do znajomych

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU41
<b>Nazwa:</b>	Przeglądanie społeczności
<b>Priorytet:</b>	Średni
<b>Aktorzy:</b>	Użytkownik zalogowany
<b>Opis:</b>	Użytkownik przegląda społeczności, grupy lub listy znajomych powiązane z aplikacją.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany.
<b>Warunki końcowe:</b>	Lista społeczności lub znajomych została wyświetlona.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik przechodzi do sekcji społeczności.</li> <li>2. System pobiera listę społeczności i znajomych użytkownika.</li> <li>3. System wyświetla listę z możliwością przechodzenia do profili i czatów.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li value="2">2a. Nie udało się pobrać danych – system wyświetla komunikat o błędzie.</li> </ol>

**Tabela 4.41:** Scenariusz przypadku użycia: Przeglądanie społeczności

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU42
<b>Nazwa:</b>	Przeglądanie dodanych spotów
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik zalogowany, Usługa do wyświetlania mapy
<b>Opis:</b>	Użytkownik przegląda listę/siatkę spotów, które sam dodał.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i znajduje się w widoku panelu użytkownika lub sekcji „Moje spedy”.
<b>Warunki końcowe:</b>	Lista dodanych spotów użytkownika została wyświetlona (np. na mapie i/lub w formie listy).
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>Użytkownik przechodzi do sekcji „Moje spedy”.</li> <li>System pobiera listę spotów dodanych przez użytkownika.</li> <li>System wyświetla listę spotów oraz znaczniki na mapie.</li> <li>Użytkownik wybiera spota, aby przejść do jego szczegółów.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>Użytkownik nie dodał jeszcze żadnego spota – system wyświetla komunikat i proponuje dodanie pierwszego spota.</li> </ol>

**Tabela 4.42:** Scenariusz przypadku użycia: Przeglądanie dodanych spotów

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU43
<b>Nazwa:</b>	Edycja danych użytkownika
<b>Priorytet:</b>	Wysoki
<b>Aktorzy:</b>	Użytkownik zalogowany

<b>Opis:</b>	Użytkownik modyfikuje swoje dane profilu (np. nazwę, opis, avatar).
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i znajduje się w widoku edycji profilu.
<b>Warunki końcowe:</b>	Zaktualizowane dane profilu są zapisane i widoczne w aplikacji.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik otwiera widok edycji profilu.</li> <li>2. Użytkownik wprowadza zmiany w danych profilu.</li> <li>3. Użytkownik zapisuje zmiany.</li> <li>4. System waliduje dane i zapisuje zaktualizowany profil.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<p>4a. Dane są niepoprawne lub niekompletne – system wyświetla komunikat o błędzie i zaznacza pola do poprawy.</p>

**Tabela 4.43:** Scenariusz przypadku użycia: Edycja danych użytkownika

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU44
<b>Nazwa:</b>	Przeglądanie dodanych zdjęć do spotów
<b>Priorytet:</b>	Średni
<b>Aktorzy:</b>	Użytkownik zalogowany, Usługa do przechowywania plików w chmurze
<b>Opis:</b>	Użytkownik przegląda wszystkie zdjęcia powiązane ze spotami, które dodał.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i znajduje się w sekcji mediów (np. „Moje zdjęcia”).

<b>Warunki końcowe:</b>	Lista lub galeria zdjęć powiązanych ze spotami użytkownika została wyświetlona.
<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik otwiera sekcję przeglądania zdjęć ze spotów.</li> <li>2. System pobiera metadane zdjęć z usługi przechowywania plików.</li> <li>3. System wyświetla galerię zdjęć z podstawowymi informacjami (np. nazwa spota, data dodania).</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>2a. Użytkownik nie dodał jeszcze zdjęć – system wyświetla informację o braku zdjęć.</li> <li>2b. Nie udało się pobrać danych – system wyświetla komunikat o błędzie.</li> </ol>

**Tabela 4.44:** Scenariusz przypadku użycia: Przeglądanie dodanych zdjęć do spotów

KARTA SCENARIUSZA PRZYPADKU UŻYCIA	
<b>Identyfikator:</b>	PU45
<b>Nazwa:</b>	Przeglądanie dodanych filmów do spotów
<b>Priorytet:</b>	Średni
<b>Aktorzy:</b>	Użytkownik zalogowany, Usługa do przechowywania plików w chmurze
<b>Opis:</b>	Użytkownik przegląda filmy powiązane ze spotami, które dał.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i znajduje się w sekcji mediów (np. „Moje filmy”).
<b>Warunki końcowe:</b>	Lista lub galeria filmów powiązanych ze spotami użytkownika została wyświetlona.

<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik otwiera sekcję przeglądania filmów ze spotów.</li> <li>2. System pobiera metadane filmów z usługi przechowywania plików.</li> <li>3. System wyświetla listę/galerię filmów z podstawowymi informacjami.</li> <li>4. Użytkownik wybiera film, aby go odtworzyć.</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li value="2">2a. Użytkownik nie dodał jeszcze filmów – system wyświetla informację o braku filmów.</li> <li value="3">3a. Nie udało się pobrać filmów – system wyświetla komunikat o błędzie.</li> </ol>

**Tabela 4.45:** Scenariusz przypadku użycia: Przeglądanie dodanych filmów do spotów

<b>KARTA SCENARIUSZA PRZYPADKU UŻYCIA</b>	
<b>Identyfikator:</b>	PU46
<b>Nazwa:</b>	Przeglądanie dodanych komentarzy do spotów
<b>Priorytet:</b>	Średni
<b>Aktorzy:</b>	Użytkownik zalogowany
<b>Opis:</b>	Użytkownik przegląda komentarze dodane do spotów, które sam utworzył.
<b>Warunki wstępne:</b>	Użytkownik jest zalogowany i otwiera sekcję komentarzy do swoich spotów.
<b>Warunki końcowe:</b>	Lista komentarzy do spotów użytkownika została wyświetlona.

<b>Główny przepływ zdarzeń:</b>	<ol style="list-style-type: none"> <li>1. Użytkownik przechodzi do sekcji komentarzy do własnych spotów.</li> <li>2. System pobiera komentarze powiązane ze spotami użytkownika.</li> <li>3. System wyświetla komentarze (np. w kolejności chronologicznej).</li> </ol>
<b>Alternatywne przepływy zdarzeń:</b>	<ol style="list-style-type: none"> <li>2a. Żaden z spotów użytkownika nie ma komentarzy – system wyświetla odpowiednią informację.</li> <li>3a. Nie udało się pobrać komentarzy – system wyświetla komunikat o błędzie.</li> </ol>

**Tabela 4.46:** Scenariusz przypadku użycia: Przeglądanie dodanych komentarzy do spotów

## 4.2 Wymagania

W niniejszym rozdziale przedstawiono wymagania stawiane projektowanemu systemowi. Celem rozdziału jest zebranie w jednym miejscu oczekiwania interesariuszy oraz usystematyzowanie ich w postaci jasno zdefiniowanych kategorii wymagań.

W pracy wyróżniono następujące typy wymagań:

- **Wymagania ogólne** – opisują system na wysokim poziomie, z perspektywy celu biznesowego i użytkownika, bez wchodzenia w szczegóły techniczne. Określają, jakie główne problemy ma rozwiązywać system i jakie korzyści ma dostarczać interesariuszom.
- **Wymagania dziedzinowe** – wynikają ze specyfiki obszaru, w którym wykorzystywany jest system (domena problemu). Odzwierciedlają reguły biznesowe, ograniczenia prawne oraz ustaloną praktykę w danej dziedzinie i muszą być spełnione niezależnie od przyjętych rozwiązań technicznych.

- **Wymagania funkcjonalne** – opisują konkretne funkcje systemu widziane z perspektywy użytkownika lub innego aktora. Definiują, jakie operacje system ma umożliwiać, jakie dane przetwarzać oraz jak reagować na określone zdarzenia i scenariusze użycia.
- **Wymagania pozafunkcjonalne** – określają, *jak* system ma realizować swoje funkcje, a nie *co* ma robić. Obejmują m.in. wymagania dotyczące wydajności, bezpieczeństwa, niezawodności, użyteczności, skalowalności oraz utrzymywalności rozwiązania.
- **Wymagania dotyczące interfejsu z otoczeniem** – opisują sposób komunikacji systemu z innymi systemami, urządzeniami lub usługami zewnętrznymi. Określają format wymienianych danych, wykorzystywane protokoły, kierunek i częstotliwość wymiany informacji oraz wymagania dotyczące integracji.
- **Wymagania dotyczące środowiska docelowego** – definiują warunki techniczne, w jakich system ma być uruchamiany i eksploatowany. Dotyczą m.in. platformy sprzętowej i programowej, systemów operacyjnych, zasobów sieciowych oraz innych elementów infrastruktury niezbędnych do poprawnego działania aplikacji. (TODO do usuniecia?)

W dalszej części rozdziału wymagania zostały logicznie pogrupowane według obszarów funkcjonalnych systemu: przedstawiono wymagania ogólne, a następnie wymagania dotyczące czatu, forum, mapy oraz panelu użytkownika.

#### 4.2.1 Wymagania ogólne

##### 4.2.1.1 Wymagania ogólne dla czatu

KARTA WYMAGANIA OGÓLNEGO DLA CZATU			
Identyfikator:	WOCZAT-01	Priorytet:	S
Nazwa:	Wysyłanie wiadomości na czacie		

KARTA WYMAGANIA OGÓLNEGO DLA CZATU (cd.)	
<b>Opis:</b>	System umożliwia użytkownikowi wysyłanie wiadomości w ramach wybranego czatu, tak aby uczestnicy mogli przekazywać sobie wiedzę na temat dronów lub umawiać się na wspólne spotkania w danym <a href="#">spocie</a> .
<b>Udziałowiec:</b>	UO3
<b>Wymagania powiązane:</b>	Wymagania funkcjonalne dla czatu: WFCZAT-01 ( <a href="#">4.1</a> ), WFCZAT-02 ( <a href="#">4.2</a> ), WFCZAT-03 ( <a href="#">4.3</a> ), WFCZAT-04 ( <a href="#">4.4</a> ), WFCZAT-06 ( <a href="#">4.6</a> ), WFCZAT-10 ( <a href="#">4.10</a> ).

**Tabela 4.1:** Karta wymagania ogólnego dla czatu: Wysyłanie wiadomości na czacie

KARTA WYMAGANIA OGÓLNEGO DLA CZATU			
Identyfikator:	WOCZAT-02	Priorytet:	S
<b>Nazwa:</b>	Edycja czatu		
<b>Opis:</b>	System umożliwia użytkownikowi wprowadzanie podstawowych zmian w konfiguracji czatu, takich jak nazwa czatu czy skład uczestników.		
<b>Udziałowiec:</b>	UO3		
<b>Wymagania powiązane:</b>	Wymagania funkcjonalne dla czatu: WFCZAT-08 ( <a href="#">4.8</a> ), WFCZAT-09 ( <a href="#">4.9</a> ), WFCZAT-11 ( <a href="#">4.11</a> ).		

**Tabela 4.2:** Karta wymagania ogólnego dla czatu: Edycja czatu

KARTA WYMAGANIA OGÓLNEGO DLA CZATU			
Identyfikator:	WOCZAT-03	Priorytet:	S
<b>Nazwa:</b>	Przeglądanie historii czatu		
<b>Opis:</b>	System udostępnia użytkownikowi możliwość przeglądania wcześniejszych wiadomości na czacie, tak aby mógł wracać do poprzednich rozmów.		
<b>Udziałowiec:</b>	UO3		

KARTA WYMAGANIA OGÓLNEGO DLA CZATU (cd.)	
<b>Wymagania powiązane:</b>	Wymagania funkcjonalne dla czatu: WFCZAT-07 ( <a href="#">4.7</a> ), WFCZAT-12 ( <a href="#">4.12</a> ).

**Tabela 4.3:** Karta wymagania ogólnego dla czatu: Przeglądanie historii czatu

KARTA WYMAGANIA OGÓLNEGO DLA CZATU			
<b>Identyfikator:</b>	WOCZAT-04	<b>Priorytet:</b>	S
<b>Nazwa:</b>	Tworzenie czatu		
<b>Opis:</b>	System umożliwia użytkownikowi inicjowanie nowych rozmów poprzez tworzenie czatów prywatnych (1:1) lub grupowych z wybranymi uczestnikami.		
<b>Udziałowiec:</b>	UO3		
<b>Wymagania powiązane:</b>	Wymaganie funkcjonalne dla czatu: WFCZAT-05 ( <a href="#">4.5</a> ).		

**Tabela 4.4:** Karta wymagania ogólnego dla czatu: Tworzenie czatu

## 4.2.2 Wymagania funkcjonalne

Niniejszy rozdział zawiera wymagania funkcjonalne postawione systemowi. Został on podzielony tematycznie.

### 4.2.2.1 Funkcjonalności dla mapy

### 4.2.2.2 Wymagania funkcjonalne dla czatu

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU			
<b>Identyfikator:</b>	WFCZAT-01	<b>Priorytet:</b>	S
<b>Nazwa:</b>	Wysyłanie GIF-ów		
<b>Opis:</b>	System umożliwia wysyłanie w wiadomościach animowanych obrazów <a href="#">GIF</a> w ramach wybranego czatu (1:1 lub grupowego).		

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU (cd.)	
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>Użytkownik może wybrać animowany obraz <a href="#">GIF</a> z wbudowanego okna wyboru i dołączyć go do wiadomości.</li> <li>Po wysłaniu <a href="#">GIF</a> wyświetla się poprawnie w treści czatu u wszystkich uczestników.</li> </ul>
<b>Dane wejściowe:</b>	Identyfikator czatu, identyfikator nadawcy, wybrany <a href="#">GIF</a> (identyfikator z usługi zewnętrznej).
<b>Warunki początkowe:</b>	Użytkownik jest zalogowany i jest członkiem danego czatu.
<b>Warunki końcowe:</b>	Wiadomość z <a href="#">GIFem</a> jest zapisana w historii czatu i widoczna dla wszystkich uprawnionych uczestników.
<b>Sytuacje wyjątkowe:</b>	Brak połączenia z serwerem, przekroczony limit rozmiaru pliku, nieobsługiwany format, błąd usługi zewnętrznej.
<b>Udziałowiec:</b>	U3
<b>Realizator:</b>	Adam Langmesser
<b>Notatka:</b>	— brak
<b>Wymagania powiązane:</b>	Wymaganie ogólne WOCZAT-01 ( <a href="#">4.1</a> ).

**Tabela 4.1:** Wymaganie funkcjonalne dla czatu: Wysyłanie GIF-ów

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU			
<b>Identyfikator:</b>	WFCZAT-02	<b>Priorytet:</b>	S
<b>Nazwa:</b>	Wysyłanie plików		
<b>Opis:</b>	System umożliwia dołączanie i wysyłanie plików (np. dokumentów, zdjęć, archiwów) jako załączników do wiadomości na czacie.		

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU (cd.)	
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>Użytkownik może wybrać jeden lub wiele plików z dysku i dołączyć je do wiadomości.</li> <li>Odbiorca widzi w czacie element reprezentujący plik.</li> <li>Kliknięcie w załącznik umożliwia pobranie lub otwarcie pliku.</li> </ul>
<b>Dane wejściowe:</b>	Identyfikator czatu, identyfikator nadawcy, co najmniej jeden plik do wysłania.
<b>Warunki początkowe:</b>	Użytkownik jest zalogowany, należy do danego czatu; plik nie przekracza ustalonego limitu rozmiaru i jest dozwolonego typu.
<b>Warunki końcowe:</b>	Wiadomość z załącznikiem jest zapisana w systemie, a plik jest dostępny do pobrania dla uczestników czatu.
<b>Sytuacje wyjątkowe:</b>	Przekroczony limit rozmiaru, nieobsługiwany typ pliku, błąd przesyłania, brak miejsca w magazynie plików.
<b>Udziałowiec:</b>	U3
<b>Realizator:</b>	Adam Langmesser
<b>Notatka:</b>	— brak
<b>Wymagania powiązane:</b>	Wymaganie ogólne WOCZAT-01 ( <a href="#">4.1</a> ).

**Tabela 4.2:** Wymaganie funkcjonalne dla czatu: Wysyłanie plików

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU			
<b>Identyfikator:</b>	WFCZAT-03	<b>Priorytet:</b>	S
<b>Nazwa:</b>	Wysyłanie wiadomości prywatnych		
<b>Opis:</b>	System umożliwia prowadzenie prywatnych rozmów 1:1 pomiędzy dwoma użytkownikami.		

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU (cd.)	
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>Użytkownik może rozpoczęć nowy czat 1:1 z innym użytkownikiem lub kontynuować istniejący.</li> <li>Wiadomości z prywatnego czatu są widoczne wyłącznie dla tych dwóch użytkowników.</li> <li>Nowe wiadomości pojawiają się w czasie zbliżonym do rzeczywistego bez konieczności przeładowania strony.</li> </ul>
<b>Dane wejściowe:</b>	Identyfikator nadawcy, identyfikator odbiorcy, treść wiadomości oraz ewentualne załączniki.
<b>Warunki początkowe:</b>	Obaj użytkownicy posiadają aktywne konta.
<b>Warunki końcowe:</b>	Wiadomości prywatne są zapisane w historii czatu 1:1 i dostępne po ponownym otwarciu rozmowy.
<b>Sytuacje wyjątkowe:</b>	Brak połączenia z serwerem, błąd walidacji treści.
<b>Udziałowiec:</b>	U3
<b>Realizator:</b>	Adam Langmesser
<b>Notatka:</b>	— brak
<b>Wymagania powiązane:</b>	Wymaganie ogólne WOCZAT-01 ( <a href="#">4.1</a> ).

**Tabela 4.3:** Wymaganie funkcjonalne dla czatu: Wysyłanie wiadomości prywatnych

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU			
<b>Identyfikator:</b>	WFCZAT-04	<b>Priorytet:</b>	S
<b>Nazwa:</b>	Wysyłanie wiadomości do wielu osób jednocześnie		
<b>Opis:</b>	System umożliwia wysyłanie jednej wiadomości do wielu użytkowników w ramach czatu grupowego.		

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU (cd.)	
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>Wiadomość wysłana w czacie grupowym jest dostarczana wszystkim jego członkom.</li> <li>UI wyraźnie wskazuje, że rozmowa to czat grupowy (np. nazwa, avatar, liczba uczestników).</li> <li>Nowi uczestnicy dołączeni do czatu widzą historię rozmowy.</li> </ul>
<b>Dane wejściowe:</b>	Identyfikator czatu grupowego, identyfikator nadawcy, treść wiadomości oraz ewentualne załączniki.
<b>Warunki początkowe:</b>	Czał typu grupowego istnieje, a użytkownik wysyłający wiadomość jest jego członkiem.
<b>Warunki końcowe:</b>	Wiadomość jest zapisana w historii czatu i widoczna dla wszystkich aktualnych uczestników.
<b>Sytuacje wyjątkowe:</b>	Brak członkostwa nadawcy w czacie, błąd autoryzacji, przekroczenie limitu użytkowników w czacie.
<b>Udziałowiec:</b>	U3
<b>Realizator:</b>	Adam Langmesser
<b>Notatka:</b>	— brak
<b>Wymagania powiązane:</b>	Wymaganie ogólne WOCZAT-01 ( <a href="#">4.1</a> ).

**Tabela 4.4:** Wymaganie funkcjonalne dla czatu: Wysyłanie wiadomości do wielu osób jednocześnie

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU			
<b>Identyfikator:</b>	WFCZAT-05	<b>Priorytet:</b>	S
<b>Nazwa:</b>	Rozpoczynanie nowego czatu		
<b>Opis:</b>	System umożliwia użytkownikowi utworzenie nowego czatu prywatnego lub grupowego oraz dodanie do niego wskazanych uczestników.		

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU (cd.)	
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>Użytkownik może zainicjować nowy czat z widoku listy czatów lub profilu innego użytkownika.</li> <li>Po wysłaniu pierwszej wiadomości przez twórcę, czat pojawia się na liście czatów wszystkich jego członków.</li> </ul>
<b>Dane wejściowe:</b>	Typ czatu (prywatny/grupowy), lista identyfikatorów uczestników, identyfikator tworzącego użytkownika.
<b>Warunki początkowe:</b>	Użytkownik jest zalogowany i ma uprawnienia do zakładania nowych czatów.
<b>Warunki końcowe:</b>	Nowy czat jest zapisany w <a href="#">bazie danych</a> , powiązany z uczestnikami i widoczny w ich listach czatów.
<b>Sytuacje wyjątkowe:</b>	Próba utworzenia czatu z nieistniejącym użytkownikiem, przekroczenie maksymalnej liczby uczestników, błąd zapisu w <a href="#">bazie danych</a> .
<b>Udziałowiec:</b>	U3
<b>Realizator:</b>	Adam Langmesser
<b>Notatka:</b>	— brak
<b>Wymagania powiązane:</b>	Wymaganie ogólne WOCZAT-04 ( <a href="#">4.4</a> ).

**Tabela 4.5:** Wymaganie funkcjonalne dla czatu: Rozpoczynanie nowego czatu

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU			
<b>Identyfikator:</b>	WFCZAT-06	<b>Priorytet:</b>	S
<b>Nazwa:</b>	Wysyłanie emotikonów		
<b>Opis:</b>	System umożliwia wysyłanie emotikonów ( <a href="#">emoji</a> ) w treści wiadomości na czacie.		
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>Użytkownik może wstawiać emotikony z wbudowanego panelu wyboru <a href="#">emoji</a>.</li> </ul>		
<b>Dane wejściowe:</b>	Identyfikator czatu, identyfikator nadawcy, treść wiadomości zawierająca <a href="#">emoji</a> .		

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU (cd.)	
<b>Warunki początkowe:</b>	Użytkownik jest zalogowany i ma dostęp do czatu.
<b>Warunki końcowe:</b>	Wiadomość z emotikonami jest zapisana w historii czatu i wyświetlana u wszystkich uczestników.
<b>Sytuacje wyjątkowe:</b>	Problemy z kodowaniem znaków, brak wsparcia dla części emoji w danej przeglądarce lub systemie.
<b>Udziałowiec:</b>	U3
<b>Realizator:</b>	Adam Langmesser
<b>Notatka:</b>	— brak
<b>Wymagania powiązane:</b>	Wymaganie ogólne WOCZAT-01 ( <a href="#">4.1</a> ).

**Tabela 4.6:** Wymaganie funkcjonalne dla czatu: Wysyłanie emotikonów

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU			
<b>Identyfikator:</b>	WFCZAT-07	<b>Priorytet:</b>	S
<b>Nazwa:</b>	Dostępność czatu po utworzeniu		
<b>Opis:</b>	Czał po utworzeniu pozostaje dostępny dla jego członków; użytkownik może później odnaleźć czat i wrócić do wcześniejszej konwersacji.		
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>• Utworzone czaty pojawiają się na liście czatów użytkownika.</li> <li>• Po ponownym zalogowaniu użytkownik może otworzyć istniejący czat i zobaczyć jego historię.</li> </ul>		
<b>Dane wejściowe:</b>	Kontekst zalogowanego użytkownika oraz parametry filtrów listy czatów.		
<b>Warunki początkowe:</b>	Czał został wcześniej utworzony, a użytkownik jest jego członkiem.		
<b>Warunki końcowe:</b>	Lista czatów użytkownika prezentuje wszystkie czaty, do których ma on dostęp.		

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU (cd.)	
<b>Sytuacje wyjątkowe:</b>	Błąd autoryzacji, błąd <a href="#">bazę danych</a> podczas pobierania listy czatów.
<b>Udziałowiec:</b>	U3
<b>Realizator:</b>	Adam Langmesser
<b>Notatka:</b>	— brak
<b>Wymagania powiązane:</b>	Wymaganie ogólne WOCZAT-03 ( <a href="#">4.3</a> ).

**Tabela 4.7:** Wymaganie funkcjonalne dla czatu: Dostępność czatu po utworzeniu

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU			
<b>Identyfikator:</b>	WFCZAT-08	<b>Priorytet:</b>	S
<b>Nazwa:</b>	Edytowanie nazwy czatu grupowego		
<b>Opis:</b>	System umożliwia zmianę nazwy istniejącego czatu grupowego przez członka czatu.		
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>• Użytkownik może edytować nazwę z poziomu ustawień czatu.</li> <li>• Nowa nazwa jest natychmiast widoczna na liście czatów u wszystkich uczestników.</li> <li>• Historia wiadomości pozostaje niezmieniona po zmianie nazwy czatu.</li> </ul>		
<b>Dane wejściowe:</b>	Identyfikator czatu grupowego, nowa nazwa czatu, identyfikator użytkownika wykonującego operację.		
<b>Warunki początkowe:</b>	Użytkownik jest członkiem czatu.		
<b>Warunki końcowe:</b>	Nazwa czatu jest zaktualizowana w <a href="#">bazie danych</a> i w interfejsie użytkownika.		
<b>Sytuacje wyjątkowe:</b>	Błąd walidacji nowej nazwy, błąd zapisu w <a href="#">bazie danych</a> .		

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU (cd.)	
<b>Udziałowiec:</b>	U3
<b>Realizator:</b>	Adam Langmesser
<b>Notatka:</b>	— brak
<b>Wymagania powiązane:</b>	Wymaganie ogólne WOCZAT-02 ( <a href="#">4.2</a> ).

**Tabela 4.8:** Wymaganie funkcjonalne dla czatu: Edytowanie nazwy czatu grupowego

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU			
<b>Identyfikator:</b>	WFCZAT-09	<b>Priorytet:</b>	S
<b>Nazwa:</b>	Edycja zdjęcia czatu grupowego		
<b>Opis:</b>	System umożliwia zmianę obrazu/avatara reprezentującego czat grupowy.		
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>• Członek czatu może wgrać nowe zdjęcie czatu grupowego.</li> <li>• Zmienione zdjęcie jest widoczne na liście czatów i w nagłówku rozmowy.</li> <li>• Niepoprawne formaty lub zbyt duże pliki są odrzucane z informacją o błędzie.</li> </ul>		
<b>Dane wejściowe:</b>	Identyfikator czatu grupowego, nowy plik graficzny, identyfikator użytkownika wykonującego operację.		
<b>Warunki początkowe:</b>	Użytkownik posiada uprawnienia do edycji ustawień czatu grupowego.		
<b>Warunki końcowe:</b>	Nowe zdjęcie czatu jest zapisane w systemie i używane w <a href="#">interfejsie użytkownika</a> .		
<b>Sytuacje wyjątkowe:</b>	Nieobsługiwany format obrazu, przekroczony limit rozmiaru, błąd przesyłania lub zapisu w magazynie plików.		
<b>Udziałowiec:</b>	U3		
<b>Realizator:</b>	Adam Langmesser		

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU (cd.)	
<b>Notatka:</b>	— brak
<b>Wymagania powiązane:</b>	Wymaganie ogólne WOCZAT-02 ( <a href="#">4.2</a> ).

**Tabela 4.9:** Wymaganie funkcjonalne dla czatu: Edycja zdjęcia czatu grupowego

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU			
<b>Identyfikator:</b>	WFCZAT-10	<b>Priorytet:</b>	W
<b>Nazwa:</b>	Edycja wysłanej wiadomości		
<b>Opis:</b>	System umożliwia użytkownikowi edycję treści wcześniej wysłanej wiadomości na czacie (np. poprawa literówki).		
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>Użytkownik może edytować swoje wiadomości przez ograniczony czas od momentu wysłania (konfigurowalny limit).</li> <li>Zmieniona wiadomość jest oznaczona etykietą „(edytowano)”.</li> <li>Odbiorcy widzą zaktualizowaną treść bez duplikowania wiadomości.</li> </ul>		
<b>Dane wejściowe:</b>	Identyfikator wiadomości, nowa treść wiadomości, identyfikator użytkownika wykonującego edycję.		
<b>Warunki początkowe:</b>	Użytkownik jest autorem wiadomości i mieści się w dozwolonym oknie czasowym edycji.		
<b>Warunki końcowe:</b>	Treść wiadomości w <a href="#">bazie danych</a> zostaje zaktualizowana, a widok czatu odświeżony u wszystkich uczestników.		
<b>Sytuacje wyjątkowe:</b>	Próba edycji cudzej wiadomości, przekroczony limit czasu, błąd validacji treści, brak połączenia z serwerem.		
<b>Udziałowiec:</b>	U3		
<b>Realizator:</b>	Adam Langmesser		
<b>Notatka:</b>	— Ze względu na konieczność priorytetyzacji pracy zespół zdecydował się zrezygnować z danego wymagania.		

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU (cd.)	
<b>Wymagania powiązane:</b>	Wymaganie ogólne WOCZAT-01 ( <a href="#">4.1</a> ).

**Tabela 4.10:** Wymaganie funkcjonalne dla czatu: Edycja wysłanej wiadomości

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU			
<b>Identyfikator:</b>	WFCZAT-11	<b>Priorytet:</b>	S
<b>Nazwa:</b>	Dodawanie użytkowników do istniejącego czatu		
<b>Opis:</b>	System umożliwia dodawanie nowych użytkowników do już istniejącego czatu grupowego.		
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>• Właściciel lub uprawniony użytkownik może wyszukać i dodać nowe osoby do czatu.</li> <li>• Nowo dodani użytkownicy pojawiają się na liście uczestników i mogą od razu brać udział w rozmowie.</li> </ul>		
<b>Dane wejściowe:</b>	Identyfikator czatu grupowego, lista identyfikatorów użytkowników do dodania, identyfikator użytkownika wykonującego operacje.		
<b>Warunki początkowe:</b>	Czał istnieje.		
<b>Warunki końcowe:</b>	Lista uczestników czatu jest zaktualizowana; nowe osoby są powiązane z czatem w <a href="#">bazie danych</a> .		
<b>Sytuacje wyjątkowe:</b>	Dodanie użytkownika, który jest już członkiem czatu; przekroczenie maksymalnej liczby uczestników.		
<b>Udziałowiec:</b>	U3		
<b>Realizator:</b>	Adam Langmesser		
<b>Notatka:</b>	— brak		
<b>Wymagania powiązane:</b>	Wymaganie ogólne WOCZAT-02 ( <a href="#">4.2</a> ).		

**Tabela 4.11:** Wymaganie funkcjonalne dla czatu: Dodawanie użytkowników do istniejącego czatu

KARTA WYMAGANIA FUNKCJONALNEGO DLA CZATU			
<b>Identyfikator:</b>	WFCZAT-12	<b>Priorytet:</b>	S
<b>Nazwa:</b>	Wyświetlanie starszych wiadomości		
<b>Opis:</b>	System powinien domyślnie wyświetlać co najmniej ostatnie 20 wiadomości w czacie, a starsze wiadomości pobierać na bieżąco podczas przewijania historii przez użytkownika.		
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>• Po wejściu na czat użytkownik widzi minimum 20 ostatnich wiadomości.</li> <li>• Przewijanie historii w górę automatycznie pobiera starsze wiadomości (mechanizm <a href="#">Infinite scroll</a>).</li> <li>• Dociąganie wiadomości nie powoduje zauważalnych opóźnień interfejsu.</li> </ul>		
<b>Dane wejściowe:</b>	Identyfikator czatu, parametry <a href="#">paginacji</a> .		
<b>Warunki początkowe:</b>	Użytkownik jest zalogowany i posiada dostęp do czatu; w <a href="#">bazie danych</a> istnieje historia rozmowy.		
<b>Warunki końcowe:</b>	Użytkownik ma możliwość przeglądania pełnej historii czatu.		
<b>Sytuacje wyjątkowe:</b>	Brak połączenia z serwerem, błąd <a href="#">paginacji</a> , przekroczenie limitu zapytań do <a href="#">API</a> .		
<b>Udziałowiec:</b>	U3		
<b>Realizator:</b>	Adam Langmesser		
<b>Notatka:</b>	— brak		
<b>Wymagania powiązane:</b>	Wymaganie ogólne WOCZAT-03 ( <a href="#">4.3</a> ).		

**Tabela 4.12:** Wymaganie funkcjonalne dla czatu: Wyświetlanie starszych wiadomości

#### 4.2.2.3 Funkcjonalności dla forum

#### 4.2.2.4 Funkcjonalności dla konta użytkownika

KARTA WYMAGANIA		
Identyfikator:	jednoznaczny symbol np. FO1, FO2 ..	Priorytet: M
Nazwa:	Profil użytkownika	
Opis:	Jako użytkownik chcę mieć dostęp do strony profilu, aby sprawdzić informacje o swoim koncie.	
Kryteria akceptacji:	Użytkownik widzi liczby: znajomych, obserwowanych i obserwujących, a także najpopularniejsze zdjęcia.	
Dane wejściowe:	Lista zdjęć oraz liczby: znajomych, obserwujących i obserwowanych.	
Warunki początkowe:	Użytkownik jest zalogowany.	
Warunki końcowe:	Wyświetlone informacje o profilu.	
Sytuacje wyjątkowe:	Błąd połączenia z API; brak danych profilu; brak uprawnień (401/403).	
Szczegóły implementacji:	Frontend: React + Tailwind; pobieranie danych profilu przez <code>@tanstack/react-query</code> i <code>axios</code> z <code>withCredentials</code> . Prezentacja w widoku profilu.	
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> ; promotor <a href="#">2.2</a> ; dronarze <a href="#">2.3</a> .	
Wymagania powiązane:		

Tabela 4.63: Profil użytkownika

KARTA WYMAGANIA		
Identyfikator:	jednoznaczny symbol np. FO1, FO2 ..	Priorytet: M
Nazwa:	Lista dodanych spotów	
Opis:	Jako użytkownik chcę sprawdzić listę spotów, które <u>dodałem</u> .	
Kryteria akceptacji:	Użytkownik widzi listę własnych dodanych spotów.	
Dane wejściowe:	Lista dodanych spotów.	
Warunki początkowe:	Użytkownik jest zalogowany.	
Warunki końcowe:	Wyświetlona lista dodanych spotów.	
Sytuacje wyjątkowe:	Brak wyników; błąd połączenia z API.	
Szczegóły implementacji:	Pobranie listy z backendu (endpoint listy własnych spotów) przez <code>react-query + axios</code> ; prezentacja listy z podstawowymi danymi.	
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> ; promotor <a href="#">2.2</a> ; dronarze <a href="#">2.3</a> .	
Wymagania powiązane:		

Tabela 4.64: Lista dodanych spotów

KARTA WYMAGANIA		
Identyfikator:	jednoznaczny symbol np. FO1, FO2 ..	Priorytet: M
Nazwa:	Dodanie spota	
Opis:	Jako użytkownik chcę mieć dostęp do formularza dodania spota.	
Kryteria akceptacji:	Użytkownik ma dostęp do formularza dodania spota i może go wysłać.	
Dane wejściowe:	Formularz dodania spota.	
Warunki początkowe:	Użytkownik jest zalogowany.	
Warunki końcowe:	Wyświetlony formularz dodania spota (po wysłaniu: zapis na backendzie).	
Sytuacje wyjątkowe:	Nieprawidłowe dane formularza; błąd połączenia z API.	
Szczegóły implementacji:	Formularz w React; walidacja przeglądarkowa; wysyłka przez axios (POST) z withCredentials.	
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> ; promotor <a href="#">2.2</a> ; droniarze <a href="#">2.3</a> .	
Wymagania powiązane:		

Tabela 4.65: Dodanie spota

KARTA WYMAGANIA		
Identyfikator:	jednoznaczny symbol np. FO1, FO2 ..	Priorytet: M
Nazwa:	Lista zdjęć	
Opis:	Jako użytkownik chcę mieć dostęp do listy zdjęć, które dodałem na forum, do komentarzy pod spodem oraz do spota.	
Kryteria akceptacji:	Użytkownik widzi listę swoich zdjęć.	
Dane wejściowe:	Lista zdjęć.	
Warunki początkowe:	Użytkownik jest zalogowany.	
Warunki końcowe:	Wyświetlona lista zdjęć.	
Sytuacje wyjątkowe:	Brak wyników; błąd połączenia z API.	
Szczegóły implementacji:	Pobranie listy zdjęć użytkownika przez <code>react-query + axios</code> ; prezentacja z miniaturami.	
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> ; promotor <a href="#">2.2</a> ; dronarze <a href="#">2.3</a> .	
Wymagania powiązane:		

Tabela 4.66: Lista zdjęć

KARTA WYMAGANIA		
Identyfikator:	jednoznaczny symbol np. FO1, FO2 ..	Priorytet: M
Nazwa:	Lista filmów	
Opis:	Jako użytkownik chcę mieć dostęp do listy filmów, które dodałem na forum, do komentarzy pod spodem oraz do spota.	
Kryteria akceptacji:	Użytkownik widzi listę swoich filmów.	
Dane wejściowe:	Lista filmów.	
Warunki początkowe:	Użytkownik jest zalogowany.	
Warunki końcowe:	Wyświetlona lista filmów.	
Sytuacje wyjątkowe:	Brak wyników; błąd połączenia z API.	
Szczegóły implementacji:	Pobranie listy filmów użytkownika przez <code>react-query + axios</code> ; prezentacja z miniaturami.	
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> ; promotor <a href="#">2.2</a> ; dronarze <a href="#">2.3</a> .	
Wymagania powiązane:		

Tabela 4.67: Lista filmów

KARTA WYMAGANIA		
Identyfikator:	jednoznaczny symbol np. FO1, FO2 ..	Priorytet: M
Nazwa:	Lista znajomych	
Opis:	Jako użytkownik chcę mieć dostęp do listy znajomych.	
Kryteria akceptacji:	Użytkownik ma dostęp do listy znajomych.	
Dane wejściowe:	Lista znajomych.	
Warunki początkowe:	Użytkownik jest zalogowany.	
Warunki końcowe:	Wyświetlona lista znajomych.	
Sytuacje wyjątkowe:	Brak wyników; błąd połączenia z API.	
Szczegóły implementacji:	Pobranie listy znajomych przez <code>react-query + axios</code> ; standardowa prezentacja listy.	
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> ; promotor <a href="#">2.2</a> ; drona- <a href="#">rze 2.3</a> .	
Wymagania powiązane:		

Tabela 4.68: Lista znajomych

KARTA WYMAGANIA		
Identyfikator:	jednoznaczny symbol np. FO1, FO2 ..	Priorytet: M
Nazwa:	Lista obserwujących	
Opis:	Jako użytkownik chcę mieć dostęp do listy obserwujących.	
Kryteria akceptacji:	Użytkownik ma dostęp do listy obserwujących.	
Dane wejściowe:	Lista obserwujących.	
Warunki początkowe:	Użytkownik jest zalogowany.	
Warunki końcowe:	Wyświetlona lista obserwujących.	
Sytuacje wyjątkowe:	Brak wyników; błąd połączenia z API.	
Szczegóły implementacji:	Pobranie listy obserwujących przez <code>react-query + axios</code> ; standardowa prezentacja listy.	
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> ; promotor <a href="#">2.2</a> ; drona- <a href="#">rze 2.3</a> .	
Wymagania powiązane:		

Tabela 4.69: Lista obserwujących

KARTA WYMAGANIA		
Identyfikator:	jednoznaczny symbol np. FO1, FO2 ..	Priorytet: M
Nazwa:	Lista obserwowanych	
Opis:	Jako użytkownik chcę mieć dostęp do listy obserwowanych.	
Kryteria akceptacji:	Użytkownik ma dostęp do listy obserwowanych.	
Dane wejściowe:	Lista obserwowanych.	
Warunki początkowe:	Użytkownik jest zalogowany.	
Warunki końcowe:	Wyświetlona lista obserwowanych.	
Sytuacje wyjątkowe:	Brak wyników; błąd połączenia z API.	
Szczegóły implementacji:	Pobranie listy obserwowanych przez <code>react-query + axios</code> ; standardowa prezentacja listy.	
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> ; promotor <a href="#">2.2</a> ; dronarze <a href="#">2.3</a> .	
Wymagania powiązane:		

Tabela 4.70: Lista obserwowanych

KARTA WYMAGANIA		
Identyfikator:	jednoznaczny symbol np. FO1, FO2 ..	Priorytet: M
Nazwa:	Lista spotów	
Opis:	Jako użytkownik chcę mieć dostęp do listy spotów, które polubiłem, odwiedziłem i planuję odwiedzić.	
Kryteria akceptacji:	Użytkownik ma dostęp do listy spotów w wymienionych kategoriach.	
Dane wejściowe:	Listy spotów: polubione, odwiedzone, planowane.	
Warunki początkowe:	Użytkownik jest zalogowany.	
Warunki końcowe:	Wyświetlone listy spotów.	
Sytuacje wyjątkowe:	Brak wyników; błąd połączenia z API.	
Szczegóły implementacji:	Pobranie list przez <code>react-query + axios</code> ; prezentacja w zakładkach/kategoriach.	
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> ; promotor <a href="#">2.2</a> ; dronarze <a href="#">2.3</a> .	
Wymagania powiązane:		

Tabela 4.71: Lista polubionych/odwiedzonych/planowanych spotów

KARTA WYMAGANIA		
Identyfikator:	jednoznaczny symbol np. FO1, FO2 ..	Priorytet: M
Nazwa:	Lista komentarzy	
Opis:	Jako użytkownik chcę mieć dostęp do listy komentarzy.	
Kryteria akceptacji:	Użytkownik ma dostęp do listy swoich komentarzy.	
Dane wejściowe:	Lista komentarzy.	
Warunki początkowe:	Użytkownik jest zalogowany.	
Warunki końcowe:	Wyświetlona lista komentarzy.	
Sytuacje wyjątkowe:	Brak wyników; błąd połączenia z API.	
Szczegóły implementacji:	Pobranie listy komentarzy użytkownika przez <code>react-query + axios</code> ; standardowa prezentacja listy.	
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> ; promotor <a href="#">2.2</a> ; dronarze <a href="#">2.3</a> .	
Wymagania powiązane:		

Tabela 4.72: Lista komentarzy

KARTA WYMAGANIA		
Identyfikator:	jednoznaczny symbol np. FO1, FO2 ..	Priorytet: M
Nazwa:	Ustawienia	
Opis:	Jako użytkownik chcę mieć możliwość zmiany danych.	
Kryteria akceptacji:	Użytkownik może edytować wybrane dane profilu i zapisać zmiany.	
Dane wejściowe:	Formularz edycji danych.	
Warunki początkowe:	Użytkownik jest zalogowany.	
Warunki końcowe:	Wyświetlony formularz edycji; po zapisie — aktualizowane dane.	
Sytuacje wyjątkowe:	Nieprawidłowe dane formularza; błąd połączenia z API.	
Szczegóły implementacji:	Formularz w React; validacja pól; wysyłka przez <code>axios</code> (PUT/PATCH) z <code>withCredentials</code> . Po sukcesie — komunikat i odświeżenie danych przez <code>react-query</code> .	
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> ; promotor <a href="#">2.2</a> ; drona-rze <a href="#">2.3</a> .	
Wymagania powiązane:		

Tabela 4.73: Ustawienia profilu

KARTA WYMAGANIA		
Identyfikator:	jednoznaczny symbol np. FO1, FO2 ..	Priorytet: M
Nazwa:	Resetowanie hasła	
Opis:	Jako użytkownik chcę mieć możliwość zresetowania hasła do swojego konta.	
Kryteria akceptacji:	Po kliknięciu w odpowiedni link użytkownik może zresetować hasło do konta.	
Dane wejściowe:	Adres e-mail użytkownika do wysłania linku resetującego.	
Warunki początkowe:	Użytkownik podał poprawny adres e-mail użyty przy rejestracji.	
Warunki końcowe:	Hasło zresetowane po przejściu całej procedury.	
Sytuacje wyjątkowe:	Niepoprawny adres e-mail; wygasły lub nieprawidłowy token resetu; błąd połączenia z API.	
Szczegóły implementacji:	Frontend: formularz „zapomniałem hasła” (POST do endpointu wysyłającego link resetu) oraz formularz ustawienia nowego hasła (POST/PATCH z tokenem). Wysyłka przez <code>axios</code> ; obsługa komunikatów o powodzeniu/błędach.	
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> ; promotor <a href="#">2.2</a> ; dronarze <a href="#">2.3</a> .	
Wymagania powiązane:		

Tabela 4.74: Resetowanie hasła

KARTA WYMAGANIA		
Identyfikator:	jednoznaczny symbol np. FO1, FO2 ..	Priorytet: M
Nazwa:	Dodawanie użytkowników do listy znajomych	
Opis:	Jako użytkownik chcę mieć możliwość dodawania innych użytkowników do listy znajomych.	
Kryteria akceptacji:	Użytkownik może dodać innego użytkownika do swojej listy znajomych.	
Dane wejściowe:	Dane użytkownika, którego chcemy dodać do znajomych.	
Warunki początkowe:	Użytkownik jest zalogowany.	
Warunki końcowe:	Znajomy dodany do listy i widoczny w profilu użytkownika.	
Sytuacje wyjątkowe:	Brak uprawnień; użytkownik już jest znajomym; błąd połączenia z API.	
Szczegóły implementacji:	Akcja wysłania zaproszenia do znajomych przez <code>axios</code> ; po akceptacji — aktualizacja listy (odświeżenie <code>react-query</code> ).	
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> ; promotor <a href="#">2.2</a> ; dronarze <a href="#">2.3</a> .	
Wymagania powiązane:		

Tabela 4.75: Dodawanie do znajomych

#### 4.2.2.5 Funkcjonalności dla logowania i rejestracji

KARTA WYMAGANIA			
Identyfikator:	FOXX	Priorytet:	M
Nazwa:	Logowanie i rejestracja		
Opis:	Jako użytkownik chcę mieć możliwość zalogowania się do aplikacji, korzystając z formularza lub poprzez konto Google lub GitHub.		
Kryteria akceptacji:	Użytkownik może zalogować się do aplikacji zarówno za pomocą standardowego formularza, jak i przy użyciu konta w serwisie Google lub GitHub.		
Dane wejściowe:	Dane użytkownika: adres e-mail, hasło; przy rejestracji dodatkowo nazwa użytkownika.		
Warunki początkowe:	Użytkownik niezalogowany.		
Warunki końcowe:	Działające formularze rejestracji i logowania oraz możliwość logowania za pomocą konta Google i GitHub.		
Sytuacje wyjątkowe:	Błędne dane logowania; przerwana lub nieudana autoryzacja u dostawcy (Google/GitHub).		
Szczegóły implementacji:	Frontend: formularze w React; wysyłka żądań przez axios z withCredentials. SSO: integracja z Google i GitHub (OAuth 2.0) z przekierowaniem i ustawniem sesji po stronie backendu (httpOnly cookie). Obsługa statusu 401 zgodnie z mechanizmem wylogowania.		
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> , promotor <a href="#">2.2</a> , dronaře <a href="#">2.3</a> .		
Wymagania powiązane:			

Tabela 4.76: Logowanie i rejestracja

#### 4.2.2.6 Funkcjonalności dla wyszukiwarki spotów

KARTA WYMAGANIA			
Identyfikator:	FOXX	Priorytet:	M
Nazwa:	Strona główna z podstawowymi filtrami		
Opis:	Jako użytkownik chcę mieć dostęp do strony głównej, która wyświetla karuzelę z najpopularniejszymi spotami oraz listę spotów, które można filtrować.		
Kryteria akceptacji:	Użytkownik widzi karuzelę najpopularniejszych miejsc. Karuzela zawiera zdjęcia, nazwę miejsca i miasto. Użytkownik może filtrować miejsca według lokalizacji (kraj, region, miasto).		
Dane wejściowe:	Lokalizacja użytkownika (kraj, region, miasto); dane z bazy spotów.		
Warunki początkowe:	Użytkownik nie musi być zalogowany.		
Warunki końcowe:	Użytkownik widzi popularne miejsca z wybranego miasta (np. Gdańsk) i może przejść do szczegółów danego miejsca.		
Sytuacje wyjątkowe:	Brak wyników dla wybranych filtrów; błąd połączenia z API.		
Szczegóły implementacji:	Frontend: React + Tailwind. Pobieranie danych przez <code>@tanstack/react-query</code> i <code>axios</code> (GET do backendu z parametrami lokalizacji). Filtry lokacji mapowane na parametry zapytania.		
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> , promotor <a href="#">2.2</a> , dronarze <a href="#">2.3</a> .		
Wymagania powiązane:			

Tabela 4.77: Strona główna — podstawowe filtry

KARTA WYMAGANIA			
Identyfikator:	FOXX	Priorytet:	M
Nazwa:	Strona główna z zaawansowanymi filtrami		
Opis:	Jako użytkownik chcę mieć dostęp do strony głównej, która wyświetla listę spotów, które można filtrować i sortować.		
Kryteria akceptacji:	Użytkownik widzi listę, którą może filtrować według miasta, tagów i oceny spota, a także sortować po ocenie i popularności.		
Dane wejściowe:	Lokalizacja użytkownika (miasto), wartości filtrów i sortowania; dane z bazy spotów.		
Warunki początkowe:	Użytkownik nie musi być zalogowany.		
Warunki końcowe:	Użytkownik widzi wyniki zgodne z zastosowanymi filtrami i sortowaniem oraz może przejść do szczegółów danego miejsca.		
Sytuacje wyjątkowe:	Brak wyników po zastosowaniu filtrów; błąd połączenia z API.		
Szczegóły implementacji:	Frontend: React + Tailwind. Pobieranie danych przez <code>@tanstack/react-query</code> i <code>axios</code> z parametrami: lokalizacja, tagi, minimalna ocena oraz kryterium sortowania.		
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> , promotor <a href="#">2.2</a> , dronarze <a href="#">2.3</a> .		
Wymagania powiązane:	SPXX		

Tabela 4.78: Strona główna — zaawansowane filtry

#### 4.2.2.7 Funkcjonalności dla motywu

KARTA WYMAGANIA			
Identyfikator:	FOXX	Priorytet:	M
Nazwa:	Ustawienia motywu		
Opis:	Jako użytkownik chcę móc zmienić motyw aplikacji.		
Kryteria akceptacji:	Dostępna jest opcja przełączenia motywu na <i>jasny</i> lub <i>ciemny</i> ; zmiana następuje bez przeładowania strony; ustawienie działa we wszystkich widokach.		
Dane wejściowe:	Preferencje użytkownika dotyczące motywu.		
Warunki początkowe:	Brak.		
Warunki końcowe:	Zmiana motywu widoczna jest natychmiast po kliknięciu przycisku.		
Sytuacje wyjątkowe:	Brak.		
Szczegóły implementacji:	Tailwind CSS z darkMode: 'class'; motyw przełączany przez dodanie/usunięcie klasy <b>dark</b> na elemencie <html>;		
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> , promotor <a href="#">2.2</a> , dronarze <a href="#">2.3</a> .		
Wymagania powiązane:			

Tabela 4.79: Ustawienia motywu (ręczna zmiana)

KARTA WYMAGANIA			
Identyfikator:	FOXX	Priorytet:	M
Nazwa:	Zapamiętywanie preferencji motywu		
Opis:	Jako użytkownik chcę, aby moja preferencja motywu była zapamiętana i przywracana przy kolejnym użyciu aplikacji.		
Kryteria akceptacji:	Wybrany motyw jest przywracany po ponownym włączeniu i odświeżeniu strony; preferencja jest zapamiętywana lokalnie w przeglądarce.		
Dane wejściowe:	Preferencje użytkownika zapisane lokalnie.		
Warunki początkowe:	FOXX dostępne.		
Warunki końcowe:	Motyw po uruchomieniu odpowiada ostatniej decyzji użytkownika.		
Sytuacje wyjątkowe:	Brak dostępu do magazynu trwałego — preferencja przechowywana w local storage.		
Szczegóły implementacji:	Zapis w <code>localStorage</code> pod kluczem <code>theme</code> ( <code>dark</code> lub <code>light</code> ); krótki skrypt umieszczony w <code>App.jsx</code> przed startem odczytuje <code>localStorage</code> i odpowiednio dodaje lub usuwa klasę <code>dark</code> na <code>&lt;html&gt;</code> (eliminuje mignięcie stylów).		
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> , promotor <a href="#">2.2</a> , drona-rze <a href="#">2.3</a> .		
Wymagania powiązane:			

Tabela 4.80: Zapamiętanie preferencji motywu

KARTA WYMAGANIA		
Identyfikator:	FOXX	Priorytet: S
Nazwa:	Przełącznik motywów w <a href="#">Sidebar</a>	
Opis:	Jako użytkownik chcę szybko zmieniać motyw bez wchodzenia w ustawienia.	
Kryteria akceptacji:	W <a href="#">Sidebar</a> dostępny jest przełącznik <i>Jasny-/Ciemny</i> ; posiada odpowiednio ikony <i>słońca/księżyca</i> ; zmiana następuje natychmiast.	
Dane wejściowe:	Bieżąca preferencja motywów.	
Warunki początkowe:	FOXX, FOXX dostępne.	
Warunki końcowe:	Motyw zmieniony; preferencja zaktualizowana.	
Sytuacje wyjątkowe:	Brak.	
Szczegóły implementacji:	Przycisk typu <i>toggle</i> wywołuje funkcję, która przełącza klasę <i>dark</i> na <i>document.documentElement</i> oraz aktualizuje <i>localStorage (theme = 'dark'   'light')</i> ; brak przeładowania strony.	
Udziałowiec:	Zespół projektowy <a href="#">2.1</a> , promotor <a href="#">2.2</a> , droniarze <a href="#">2.3</a> .	
Wymagania powiązane:		

Tabela 4.81: Szybki przełącznik motywów w interfejsie

#### 4.2.3 Wymagania pozafunkcjonalne

Niniejszy rozdział zawiera wymagania pozafunkcjonalne postawione systemowi. Został on podzielony tematycznie.

##### 4.2.3.1 Wymagania pozafunkcjonalne dla czatu

KARTA WYMAGANIA POZAFUNKCJONALNEGO DLA CZATU		
Identyfikator:	WPCZAT-01	Priorytet: M
Nazwa:	Ograniczenie widoczności czatów do członków	
Typ:	Bezpieczeństwo	

<b>KARTA WYMAGANIA POZAFUNKCJONALNEGO DLA CZATU (cd.)</b>	
<b>Opis:</b>	System zapewnia, że użytkownik widzi wyłącznie listę czatów oraz wiadomości z czatów, których jest członkiem. Informacje o innych czatach nie są prezentowane w interfejsie ani dostępne poprzez API.
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>• Lista czatów dla zalogowanego użytkownika zawiera wyłącznie czaty, w których jest on uczestnikiem.</li> <li>• Próba otwarcia czatu, którego użytkownik nie jest członkiem, kończy się czytelnym błędem (np. <i>brak uprawnień</i> lub <i>nie znaleziono</i>).</li> <li>• Wiadomości z czatów, do których użytkownik nie ma dostępu, nie są zwracane przez API ani widoczne w logach klienta.</li> </ul>
<b>Dane wejściowe:</b>	Kontekst zalogowanego użytkownika, identyfikator użytkownika, identyfikator czatu (dla widoku szczegółowego).
<b>Warunki początkowe:</b>	Użytkownik jest poprawnie uwierzytelniony; w systemie istnieją zarejestrowane czaty i przypisania użytkownik-czat.
<b>Warunki końcowe:</b>	Użytkownik ma dostęp wyłącznie do własnych czatów i ich wiadomości, zgodnie z zapisanymi członkostwami.
<b>Sytuacje wyjątkowe:</b>	Błędna konfiguracja uprawnień, niespójne dane o członkostwach w bazie, awaria usługi autoryzacji.
<b>Szczegóły implementacji:</b>	Filtrowanie danych po stronie backendu na podstawie identyfikatora użytkownika; wymuszenie sprawdzania członkostwa przy każdym zapytaniu o czat lub wiadomości; testy integracyjne dla scenariuszy <i>brak uprawnień</i> .
<b>Udziałowiec:</b>	Użytkownik zalogowany, administrator systemu.
<b>Realizator:</b>	Adam Langmesser

KARTA WYMAGANIA POZAFUNKCJONALNEGO DLA CZATU (cd.)	
Status:	Zrealizowano

**Tabela 4.1:** Wymaganie pozafunkcjonalne dla czatu: Ograniczenie widoczności czatów do członków

KARTA WYMAGANIA POZAFUNKCJONALNEGO DLA CZATU		
<b>Identyfikator:</b>	WPCZAT-02	<b>Priorytet:</b> M
<b>Nazwa:</b>	Wymóg zalogowania do korzystania z czatu	
<b>Typ:</b>	Bezpieczeństwo	
<b>Opis:</b>	Dostęp do funkcji czatu (lista czatów, wysyłanie i odbieranie wiadomości, tworzenie czatów) wymaga wcześniejszego zalogowania się do systemu. Użytkownik niezalogowany nie może przeglądać ani modyfikować danych czatu.	
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>• Wejście na widok czatu przez użytkownika niezalogowanego powoduje przekierowanie na stronę logowania lub komunikat o braku uprawnień.</li> <li>• Zapytania do API czatu bez ważnego tokena/autoryzacji są odrzucane.</li> <li>• Po poprawnym zalogowaniu użytkownik uzyskuje pełny dostęp do swoich czatów bez konieczności ponownego logowania w tej sesji.</li> </ul>	
<b>Dane wejściowe:</b>	Dane logowania użytkownika, token sesji lub inny mechanizm uwierzytelniania.	
<b>Warunki początkowe:</b>	Użytkownik posiada aktywne konto w systemie.	
<b>Warunki końcowe:</b>	Tylko użytkownicy zalogowani mogą korzystać z funkcji czatu; użytkownicy niezalogowani widzą jedynie ekran logowania/rejestracji.	

KARTA WYMAGANIA POZAFUNKCJONALNEGO DLA CZATU (cd.)	
Sytuacje wyjątkowe:	Wygaśnięcie sesji, utrata tokena, błąd integracji z modelem logowania.
Szczegóły implementacji:	Middleware/autoryzacja na poziomie backendu wymuszająca uwierzytelnienie dla wszystkich endpointów czatu; przechowywanie danych sesji w bezpieczny sposób (np. token JWT).
Udziałowiec:	Użytkownik zalogowany, administrator systemu.
Realizator:	Adam Langmesser
Status:	Zrealizowano

**Tabela 4.2:** Wymaganie pozafunkcjonalne dla czatu: Wymóg zalogowania do korzystania z czatu

KARTA WYMAGANIA POZAFUNKCJONALNEGO DLA CZATU			
Identyfikator:	WPCZAT-03	Priorytet:	M
Nazwa:	Grupowanie wiadomości według daty wysłania		
Typ:	Użyteczność		
Opis:	Wiadomości na czacie są prezentowane w logicznych grupach odpowiadających datom ich wysłania (np. separatory dzienne), co ułatwia użytkownikom orientację w historii rozmowy.		
Kryteria akceptacji:	<ul style="list-style-type: none"> <li>• W widoku czatu pojawiają się wizualne znaczniki dat (np. „Dzisiaj”, „Wczoraj”, konkretna data).</li> <li>• Wiadomości są zawsze przypisane do poprawnej grupy daty wysłania niezależnie od strefy czasowej klienta.</li> <li>• Zmiana zakresu historii (scrollowanie, przeładowanie) zachowuje poprawne grupowanie dat.</li> </ul>		

KARTA WYMAGANIA POZAFUNKCJONALNEGO DLA CZATU (cd.)	
<b>Dane wejściowe:</b>	Wiadomości danego czatu wraz z ich znacznikami czasu wysłania.
<b>Warunki początkowe:</b>	Wiadomości posiadają poprawnie zapisany czas wysłania w spójnym formacie.
<b>Warunki końcowe:</b>	Użytkownik widzi wiadomości zgrupowane według dat, co poprawia czytelność dłuższych rozmów.
<b>Sytuacje wyjątkowe:</b>	Błędne strefy czasowe lub niepoprawne wartości znaczników czasu w bazie danych.
<b>Szczegóły implementacji:</b>	Normalizacja dat i czasów do strefy referencyjnej (np. UTC) po stronie backendu oraz odpowiednie formatowanie i grupowanie po stronie klienta.
<b>Udziałowiec:</b>	Użytkownik zalogowany.
<b>Realizator:</b>	Adam Langmesser
<b>Status:</b>	Zrealizowano

**Tabela 4.3:** Wymaganie pozafunkcjonalne dla czatu: Grupowanie wiadomości według daty wysłania

KARTA WYMAGANIA POZAFUNKCJONALNEGO DLA CZATU			
<b>Identyfikator:</b>	WPCZAT-04	<b>Priorytet:</b>	M
<b>Nazwa:</b>	Wyraźne oznaczenie nadawcy i czasu wysłania		
<b>Typ:</b>	Użyteczność		
<b>Opis:</b>	Każda wiadomość na czacie jest opatrzona wyraźną informacją, kto jest jej nadawcą oraz kiedy została wysłana (data i czas). Informacje te są łatwo zauważalne i spójne wizualnie.		

<b>KARTA WYMAGANIA POZAFUNKCJONALNEGO DLA CZATU (cd.)</b>	
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>• Przy każdej wiadomości widoczna jest nazwa lub alias nadawcy.</li> <li>• Czas wysłania jest prezentowany w czytelnej formie (np. „12:34”, „wczoraj 21:10”) z uwzględnieniem lokalnej strefy czasowej użytkownika.</li> <li>• Przejście kursorem lub tapnięcie umożliwia wyświetlenie pełnej daty i czasu w bardziej szczegółowym formacie.</li> </ul>
<b>Dane wejściowe:</b>	Dane użytkownika–nadawcy oraz znacznik czasu wysłania każdej wiadomości.
<b>Warunki początkowe:</b>	Dane użytkowników i wiadomości są spójne; każda wiadomość ma powiązanego nadawcę i poprawny czas wysłania.
<b>Warunki końcowe:</b>	Użytkownik może jednoznacznie zidentyfikować nadawcę i czas wysłania każdej wiadomości.
<b>Sytuacje wyjątkowe:</b>	Brak danych o nadawcy lub czasie (stare wiadomości, dane testowe), błędy migracji danych.
<b>Szczegóły implementacji:</b>	Powiązanie wiadomości z tabelą użytkowników po klu-czu obcym; formatowanie czasu po stronie klienta; spójny komponent UI dla nagłówka wiadomości.
<b>Udziałowiec:</b>	Użytkownik zalogowany, moderatorzy czatu.
<b>Realizator:</b>	Adam Langmesser
<b>Status:</b>	Zrealizowano

**Tabela 4.4:** Wymaganie pozafunkcjonalne dla czatu: Wyraźne oznaczenie nadawcy i czasu wysłania

<b>KARTA WYMAGANIA POZAFUNKCJONALNEGO DLA CZATU</b>	
<b>Identyfikator:</b>	WPCZAT-05

<b>KARTA WYMAGANIA POZAFUNKCJONALNEGO DLA CZATU (cd.)</b>	
<b>Nazwa:</b>	Czas załadowania starszych wiadomości poniżej 3 sekund
<b>Typ:</b>	Wydajność
<b>Opis:</b>	Podczas przewijania historii czatu załadowanie kolejnej partii starszych wiadomości powinno trwać krócej niż 3 sekundy w typowych warunkach sieciowych.
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>• W co najmniej 95% pomiarów laboratoryjnych i testów akceptacyjnych czas pobrania starszych wiadomości mieści się w przedziale 0–3 s.</li> <li>• Interfejs wyraźnie sygnalizuje trwające ładowanie (np. animacja „ładowanie”), a po zakończeniu przewijanie jest płynne.</li> <li>• Brak zauważalnych „zawieszeń” interfejsu podczas dociągania danych.</li> </ul>
<b>Dane wejściowe:</b>	Identyfikator czatu, parametry paginacji (np. kursor, znacznik czasu), liczba wiadomości do pobrania.
<b>Warunki początkowe:</b>	W bazie istnieje historia rozmowy; serwer i baza danych działają poprawnie.
<b>Warunki końcowe:</b>	Starsze wiadomości są dołączone do historii czatu użytkownika w czasie krótszym niż 3 s w typowych warunkach.
<b>Sytuacje wyjątkowe:</b>	Bardzo słabe łącze użytkownika, awaria sieci, wysoka chwilowa niedostępność bazy danych.
<b>Szczegóły implementacji:</b>	Indeksy po znaczniku czasu w bazie danych, paginacja kursorowa, cache po stronie serwera i klienta; ograniczenie liczby pobieranych rekordów w jednym żądaniu.
<b>Udziałowiec:</b>	Użytkownik zalogowany.
<b>Realizator:</b>	Adam Langmesser
<b>Status:</b>	Zrealizowano

**Tabela 4.5:** Wymaganie pozafunkcjonalne dla czatu: Czas załadowania starszych wiadomości poniżej 3 sekund

KARTA WYMAGANIA POZAFUNKCJONALNEGO DLA CZATU		
<b>Identyfikator:</b>	WPCZAT-06	<b>Priorytet:</b> M
<b>Nazwa:</b>	Natychmiastowe wysyłanie wiadomości	
<b>Typ:</b>	Wydajność	
<b>Opis:</b>	Po wysłaniu wiadomości przez użytkownika powinna ona pojawić się w widoku czatu w czasie subiektywnie natychmiastowym (rzędu setek milisekund), a pozostały uczestnicy powinni ją zobaczyć w czasie zbliżonym do rzeczywistego.	
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>• W typowych warunkach sieciowych użytkownik widzi swoją nową wiadomość w czacie w czasie poniżej 1 s od wysłania.</li> <li>• Pozostali uczestnicy czatu otrzymują wiadomość bez konieczności ręcznego odświeżania (np. przez WebSocket).</li> <li>• W przypadku chwilowego opóźnienia interfejs sygnalizuje status wysyłania (np. ikona „wysyłanie...”, „niedostarczona”).</li> </ul>	
<b>Dane wejściowe:</b>	Identyfikator czatu, identyfikator nadawcy, treść wiadomości oraz ewentualne załączniki.	
<b>Warunki początkowe:</b>	Użytkownik jest zalogowany, posiada dostęp do czatu, serwer jest dostępny.	
<b>Warunki końcowe:</b>	Wiadomość jest zapisana w bazie danych oraz dostarczona wszystkim uprawnionym uczestnikom czatu.	
<b>Sytuacje wyjątkowe:</b>	Zerwanie połączenia sieciowego, chwilowa niedostępność serwera, przeciążenie systemu.	

KARTA WYMAGANIA POZAFUNKCJONALNEGO DLA CZATU (cd.)	
<b>Szczegóły implementacji:</b>	Wykorzystanie połączeń WebSocket lub innego kanału push; kolejka wiadomości po stronie serwera; asynchroniczne zapisy do bazy danych.
<b>Udziałowiec:</b>	Użytkownik zalogowany.
<b>Realizator:</b>	Adam Langmesser
<b>Status:</b>	Zrealizowano

**Tabela 4.6:** Wymaganie pozafunkcjonalne dla czatu: Natychmiastowe wysyłanie wiadomości

KARTA WYMAGANIA POZAFUNKCJONALNEGO DLA CZATU		
<b>Identyfikator:</b>	WPCZAT-07	<b>Priorytet:</b> M
<b>Nazwa:</b>	Zachowanie wiadomości przy chwilowej utracie połączenia	
<b>Typ:</b>	Niezawodność	
<b>Opis:</b>	W przypadku krótkotrwałej utraty połączenia sieciowego wiadomości wysłane przez użytkownika nie powinny zostać utracone: zostaną ponownie wysłane po odzyskaniu łączności lub jednoznacznie oznaczone jako niedostarczone.	
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>• Wiadomości wysłane w momencie utraty połączenia są buforowane lokalnie do czasu ponownej próby wysłania.</li> <li>• Po odzyskaniu połączenia następuje automatyczna próba ponownego dostarczenia buforowanych wiadomości.</li> <li>• W przypadku braku możliwości dostarczenia użytkownik otrzymuje czytelną informację (status „nie-dostarczona”) i może spróbować ponownie.</li> </ul>	

KARTA WYMAGANIA POZAFUNKCJONALNEGO DLA CZATU (cd.)	
<b>Dane wejściowe:</b>	Bufor lokalny wiadomości po stronie klienta, status połączenia sieciowego, identyfikator czatu i nadawcy.
<b>Warunki początkowe:</b>	Użytkownik jest zalogowany; przed utratą połączenia czat działał poprawnie.
<b>Warunki końcowe:</b>	Wiadomości wysłane w okresie chwilowej utraty połączenia są ostatecznie dostarczone lub jasno oznaczone jako niedostarczone, bez „cichej” utraty danych.
<b>Sytuacje wyjątkowe:</b>	Długotrwały brak połączenia, ręczne zamknięcie aplikacji przed synchronizacją bufora, usunięcie danych lokalnych.
<b>Szczegóły implementacji:</b>	Bufor wysyłanych wiadomości w pamięci lub local storage po stronie klienta; mechanizm ponawiania wysyłki; idempotentne mutacje po stronie backendu.
<b>Udziałowiec:</b>	Użytkownik zalogowany.
<b>Realizator:</b>	Adam Langmesser
<b>Status:</b>	Zrealizowano

**Tabela 4.7:** Wymaganie pozafunkcjonalne dla czatu: Zachowanie wiadomości przy chwilowej utracie połączenia

KARTA WYMAGANIA POZAFUNKCJONALNEGO DLA CZATU		
<b>Identyfikator:</b>	WPCZAT-08	<b>Priorytet:</b> M
<b>Nazwa:</b>	Limit wysyłanych wiadomości w jednostce czasu	
<b>Typ:</b>	Niezawodność	
<b>Opis:</b>	System ogranicza maksymalną liczbę wiadomości wysyłanych przez pojedynczego użytkownika w określonej jednostce czasu, aby chronić czat przed spamem oraz przeciążeniem.	

KARTA WYMAGANIA POZAFUNKCJONALNEGO DLA CZATU (cd.)	
<b>Kryteria akceptacji:</b>	<ul style="list-style-type: none"> <li>Przy zbyt częstym wysyłaniu wiadomości użytkownik otrzymuje informację o osiągnięciu limitu, a kolejne wiadomości są tymczasowo blokowane.</li> <li>Normalna praca czatu nie jest utrudniona dla typowych wzorców użycia (pisanie „normalnym tempem” nie wyczerpuje limitu).</li> <li>Limity mogą być konfigurowane (np. inne dla zwykłych użytkowników i administratorów).</li> </ul>
<b>Dane wejściowe:</b>	Identyfikator użytkownika, znaczniki czasu wysłanych wiadomości w ostatnim przedziale czasowym.
<b>Warunki początkowe:</b>	Użytkownik jest zalogowany; moduł czatu rejestruje czas wysłania każdej wiadomości.
<b>Warunki końcowe:</b>	Nadmierne tempo wysyłania wiadomości przez jednego użytkownika jest ograniczane, co zmniejsza ryzyko przeciążenia systemu i spamu.
<b>Sytuacje wyjątkowe:</b>	Błędy konfiguracji limitów, ataki rozproszone z wielu kont, duże wahania ruchu w krótkim czasie.
<b>Szczegóły implementacji:</b>	Mechanizm <i>rate limiting</i> na poziomie backendu (np. licznik w pamięci lub magazynie typu Redis) powiązany z identyfikatorem użytkownika lub adresu IP; zwracanie odpowiedniego kodu błędu przy przekroczeniu limitu.
<b>Udziałowiec:</b>	Użytkownik zalogowany, administrator systemu.
<b>Realizator:</b>	Adam Langmesser
<b>Status:</b>	Zrealizowano

**Tabela 4.8:** Wymaganie pozafunkcjonalne dla czatu: Limit wysyłanych wiadomości w jednostce czasu

# Rozdział 5

## Projekt

### 5.1 Wzorce projektowe

### 5.2 Architektura systemu

W niniejszym rozdziale przedstawiona zostanie architektura systemu, czyli sposób, w jaki poszczególne komponenty komunikują się między sobą, a także technologie, za pomocą których zostały stworzone.

Jednym z kluczowych etapów realizacji projektu był wybór odpowiedniej architektury systemowej. Ostatecznie przyjęto oddzielenie poszczególnych warstw aplikacji, co zapewnia większą elastyczność, skalowalność oraz ułatwia rozwój w przyszłości. Przyjęte komponenty prezentują się następująco:

- frontend – React z wykorzystaniem [TypeScriptu](#),
- backend – Java Spring Boot,
- [baza danych](#) – PostgreSQL,
- [redis](#) – wykorzystywany jako [baza danych](#) klucz-wartość pełniąca rolę warstwy [cache](#).

Jest to podejście, w którym zespół projektowy posiada największe doświadczenie, dlatego zostało ono zastosowane. Pozwala ono również na tworzenie aplikacji responsywnej, dostępnej zarówno na komputerach, jak i urządzeniach mobilnych.

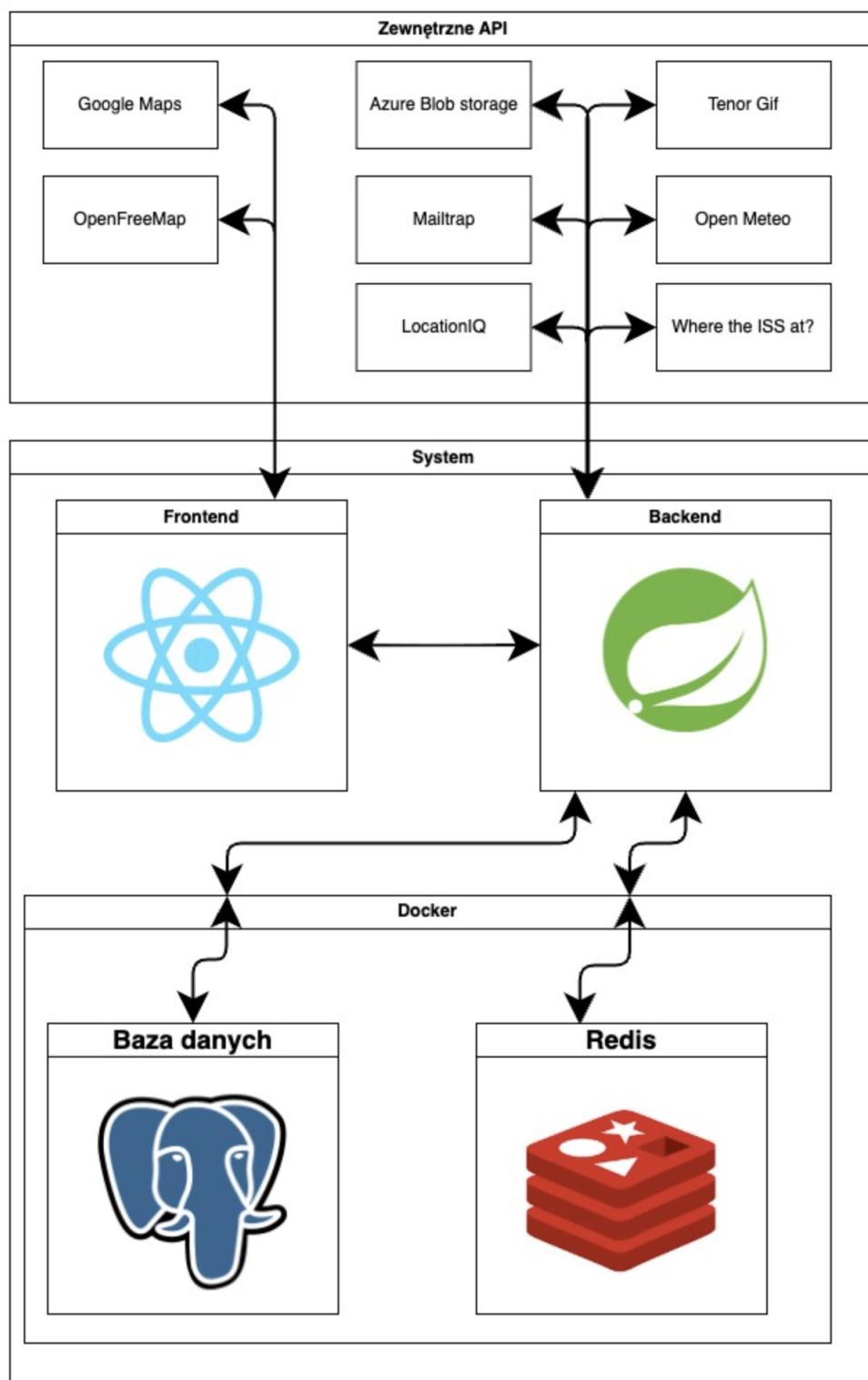
Warstwa wizualna została przygotowana przy użyciu [React](#) w wersji z [TypeScriptem](#) oraz [biblioteki](#) Tailwind CSS, zapewniającej szybkie i wygodne stylowanie komponentów. Z kolei za komunikację oraz logikę biznesową odpowiada [backend](#) oparty na [frameworku](#) Spring Boot, realizujący założenia architektury [REST API](#). Jako system zarządzania danymi wybrano relacyjną bazę danych PostgreSQL, z którą zespół posiada największe doświadczenie. Relacyjny model danych doskonale sprawdza się w tym projekcie, zapewniając integralność danych, możliwość tworzenia złożonych zapytań oraz wysoką stabilność.

[Redis](#) został wykorzystany jako warstwa [cache](#), której zadaniem jest przyspieszenie działania aplikacji poprzez ograniczenie liczby odwołań do głównej [bazy danych](#). Dzięki przechowywaniu często wykorzystywanych danych w pamięci operacyjnej znacznie skraca się czas odpowiedzi systemu, co pozytywnie wpływa na wydajność oraz skalowalność rozwiązania. Zastosowanie [Redisa](#) okazało się szczególnie korzystne w przypadku operacji powtarzalnych i odczytowych, które nie wymagają każdorazowego dostępu do relacyjnej [bazy danych](#).

### 5.2.1 Diagram architektury

Projekt aplikacji oparto na architekturze klient–serwer z podziałem na [frontend](#) i [backend](#). Takie podejście ułatwia rozwój i utrzymanie systemu oraz umożliwia skalowanie poszczególnych komponentów niezależnie od siebie. Komunikacja między [frontendem](#) a [backendem](#) odbywa się za pomocą [REST API](#), przy czym dane przesyłane są w formacie JSON. Integracja między warstwami aplikacji jest dzięki temu lekka, czytelna i łatwa do rozszerzenia w przyszłości.

Architektura została opracowana dla środowiska deweloperskiego. W obecnym zakresie prac nie uwzględniono implementacji środowiska produkcyjnego.



Rysunek 5.1: Diagram architektury

### 5.2.2 Komponenty systemu

System składa się z kilku głównych komponentów, z których każdy pełni ściśle określona rolę.

- **Frontend** – odpowiada za warstwę prezentacji oraz interfejs użytkownika dostępny dla wszystkich użytkowników systemu,
- **Backend** – odpowiada za autoryzację użytkowników oraz obsługę komunikacji między **frontendem** a **bazą danych**,
- **Baza danych** – przechowuje wszystkie dane aplikacji, w tym dane użytkowników, dane operacyjne oraz informacje potrzebne do działania systemu.
- **Redis** – wykorzystywany jako warstwa cache, przechowująca często odczytywane dane w pamięci operacyjnej, co znacząco przyspiesza działanie systemu oraz zmniejsza obciążenie głównej bazy danych.

Szczegółowy wykaz wykorzystywanych zewnętrznych API zamieszczono w rozdziale 3.

- Azure Blob Storage – [3.2](#)
- Mailtrap – [3.3](#)
- LocationIQ – [3.4](#)
- Google Maps – [3.5](#)
- OpenFreeMap – [3.6](#)
- Open Meteo – [3.7](#)
- Tenor Gif – [3.8](#)
- Where the ISS at? – [3.9](#)

## **5.3 Projekt bazy danych**

**5.3.1 Model danych**

**5.3.2 Diagram ERD**

## **5.4 Architektura interfejsu użytkownika**

**5.4.1 Projekt strony głównej**

**5.4.2 Projekt panelu logowania**

**5.4.3 Projekt mapy**

**5.4.4 Projekt chatu**

**5.4.5 Projekt forum**

**5.4.6 Projekt konta użytkownika**

## Rozdział 6

### Przebieg realizacji projektu

6.1 Sprint 1

6.2 Sprint 2

# Rozdział 7

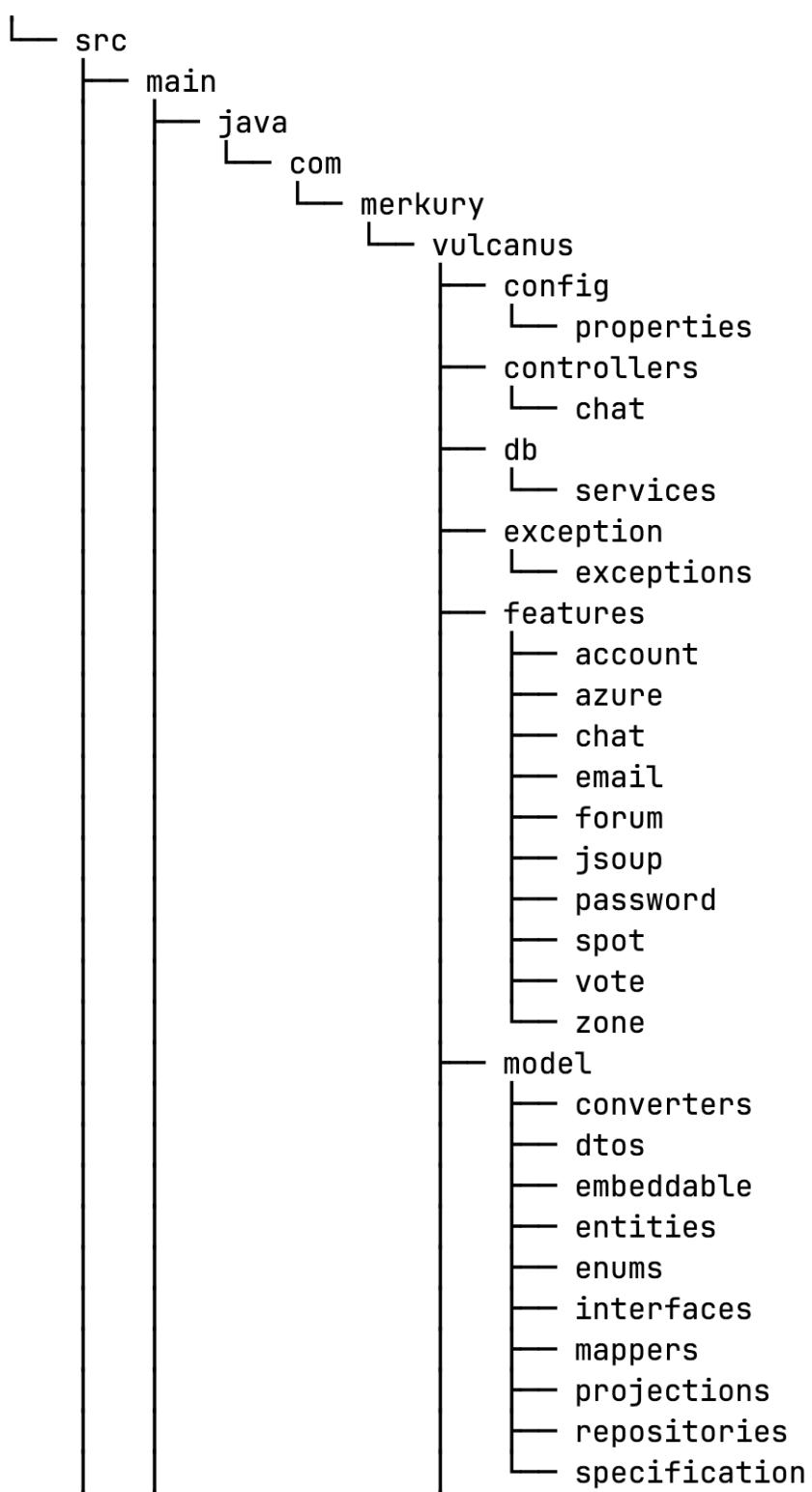
## Realizacja Projektu

### 7.1 Implementacja backendu

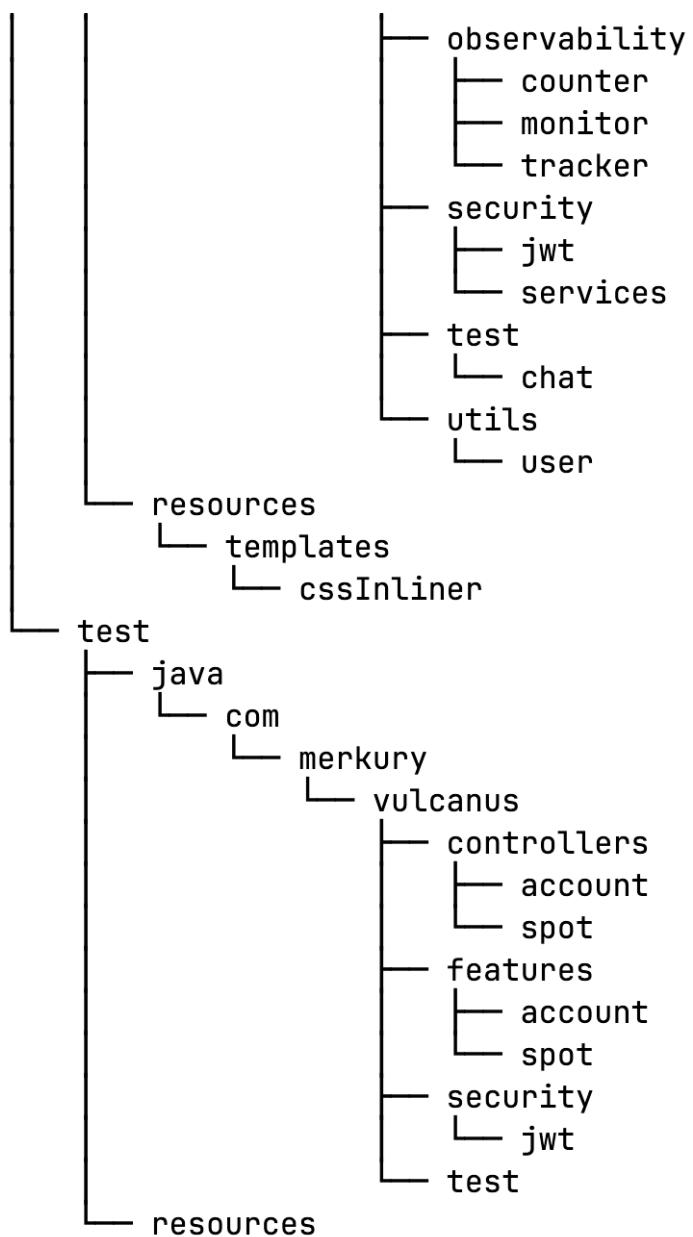
W niniejszym rozdziale przedstawiono strukturę backendu aplikacji, opis implementowanych endpointów, integrację z bazą danych, mechanizmy uwierzytelniania oraz proces konteneryzacji systemu.

#### 7.1.1 Struktura projektu

Backend aplikacji został zaimplementowany przy użyciu frameworka Spring Boot, co umożliwiło stworzenie spójnej i skalowej architektury w prosty sposób. W projekcie zastosowano rozwiązanie typu REST API, gdyż zespół projektowy dysponuje największym doświadczeniem w jego wykorzystaniu. Struktura projektu została zorganizowana zgodnie z podejściem folder by type, dzięki czemu każdy plik znajduje się w odpowiadającym mu katalogu. Takie podejście ułatwia zarówno lokalizację istniejących plików, jak i określenie miejsca tworzenia nowych komponentów. Poniżej przedstawiono przykładową strukturę katalogów backendu:



Rysunek 7.1: Struktura katalogów (1)



Rysunek 7.2: Struktura katalogów (2)

Dzięki takiej organizacji kod jest bardziej czytelny i łatwiejszy w utrzymaniu. Umożliwia również szybkie odnalezienie odpowiednich modułów oraz ułatwia rozbudowę projektu w przyszłości.

### 7.1.2 Endpointy systemu

Projektowany system udostępnia REST-owe API HTTP, za pomocą którego klienci komunikują się z serwerem.

Na potrzeby niniejszej pracy przez *endpoint REST API* rozumiany będzie konkretny punkt dostępu do systemu, zdefiniowany jako para:

metoda HTTP + ścieżka URL

pod którym aplikacja udostępnia określoną funkcjonalność lub zasób. Przykładowo, endpoint `GET /public/spot/{spotId}` służy do pobierania informacji o wybranym specie.

W dalszej części rozdziału przedstawiono zestawienie wszystkich endpointów HTTP systemu, a następnie szczegółowe karty wybranych endpointów, opisujące m.in. parametry wejściowe, `Query params` oraz strukturę odpowiedzi.

Zestawienie wszystkich endpointów HTTP panelu użytkownika	
Metoda HTTP	Ścieżka
GET	/user-dashboard/profile
GET	/public/user-dashboard/profile/{targetUsername}
PATCH	/user-dashboard/profile
GET	/user-dashboard/friends
GET	/public/user-dashboard/friends/{targetUsername}
PATCH	/user-dashboard/friends
PATCH	/user-dashboard/friends/change-status
GET	/user-dashboard/followers
GET	/public/user-dashboard/followers/{targetUsername}
GET	/user-dashboard/followed
GET	/public/user-dashboard/followed/{targetUsername}

GET	/user-dashboard/friends/find
GET	/user-dashboard/friends/invites
PATCH	/user-dashboard/followed
GET	/user-dashboard/favorite-spots
PATCH	/user-dashboard/favorite-spots
POST	/user-dashboard/add-spot-media
GET	/user-dashboard/is-spot-favourite
GET	/user-dashboard/photos
GET	/user-dashboard/comments
PATCH	/user-dashboard/settings
GET	/user-dashboard/settings
GET	/user-dashboard/movies
GET	/user-dashboard/photos/{targetUsername}
GET	/user-dashboard/add-spot
POST	/user-dashboard/add-spot
GET	/user-dashboard/add-spot/coordinates

**Tabela 7.1:** Zestawienie endpointów: panelu użytkownika

Zestawienie wszystkich endpointów HTTP modułu spotów	
Metoda HTTP	Ścieżka
GET	/public/spot/gallery
GET	/public/spot/gallery-media-position
GET	/public/spot/gallery-fullscreen-media
GET	/public/spot/current-view

GET	/public/spot/current-view/spot-names
GET	/public/spot/{spotId}
PATCH	/public/spot/increase-view-count
GET	/public/spot/search/map
GET	/public/spot/search/list
GET	/public/spot/names
GET	/public/spot/most-popular
GET	/public/spot/search/home-page
GET	/public/spot/search/home-page/locations
GET	/public/spot/search/home-page/advance
GET	/public/spot/get-spot-basic-weather
GET	/public/spot/get-spot-detailed-weather
GET	/public/spot/get-spot-wind-speeds
GET	/public/spot/get-spot-weather-timeline-plot-data
PATCH	/public/spot/increase-spot-media-views-count
PATCH	/public/spot/edit-spot-media-likes
GET	/spot/check-is-spot-media-liked
GET	/public/spot/get-spot-time-zone

**Tabela 7.2:** Zestawienie endpointów: modułu spotów

Zestawienie wszystkich endpointów HTTP komentarzy do spotów	
Metoda HTTP	Ścieżka
GET	/public/spot/{spotId}/comments
GET	/public/spot/{spotId}/comments/{commentId}

POST	/spot/{spotId}/comments
DELETE	/spot/comments/{commentId}
PATCH	/spot/comments/{commentId}
PATCH	/spot/comments/{commentId}/vote
GET	/spot/comments/vote-type

**Tabela 7.3:** Zestawienie endpointów: komentarzy do spotów

Zestawienie wszystkich endpointów HTTP postów forum	
Metoda HTTP	Ścieżka
GET	/public/post/{postId}
GET	/public/post
POST	/post
DELETE	/post/{postId}
PATCH	/post/{postId}
PATCH	/post/{postId}/vote
PATCH	/post/{postId}/follow
PATCH	/post/{postId}/report
GET	/public/categories-tags

**Tabela 7.4:** Zestawienie endpointów: postów forum

Zestawienie wszystkich endpointów HTTP komentarzy forum	
Metoda HTTP	Ścieżka

GET	/public/post/{postId}/comments
GET	/public/comments/{commentId}/replies
POST	/post/{postId}/comments
DELETE	/post/comments/{commentId}
PATCH	/post/comments/{commentId}
PATCH	/post/comments/{commentId}/vote
PATCH	/post/comments/{commentId}/report
POST	/comments/{commentId}/replies

**Tabela 7.5:** Zestawienie endpointów: komentarzy forum

Zestawienie wszystkich endpointów HTTP konta użytkownika i autoryzacji	
Metoda HTTP	Ścieżka
POST	/public/account/register
POST	/public/account/login
GET	/account/login-success
POST	/public/account/forgot-password
POST	/public/account/set-new-password
GET	/account/check

**Tabela 7.6:** Zestawienie endpointów: konta użytkownika i autoryzacji

Zestawienie wszystkich endpointów HTTP integracji GIF-ów (Tenor)

Metoda HTTP	Ścieżka
GET	/gifs/trending
GET	/gifs/search

**Tabela 7.7:** Zestawienie endpointów: integracji GIF-ów

Zestawienie wszystkich endpointów HTTP modułu czatu	
Metoda HTTP	Ścieżka
GET	/chats/{chatId}/messages
GET	/chats/user-chats
POST	/chats/get-or-create-private-chat
POST	/chats/{chatId}/send-files
POST	/chats/create/group
PATCH	/chats/{chatId}
GET	/chats/group-chat/add/search/{chatId}
PUT	/chats/add/users/{chatId}

**Tabela 7.8:** Zestawienie endpointów: modułu czatu

### Panel użytkownika

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP01
<b>Ścieżka:</b>	/public/user-dashboard/profile/{targetUsername}
<b>Nazwa:</b>	Pobierz profil innego użytkownika (widok publiczny)

<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>targetUsername</b>: String (nazwa użytkownika w ścieżce URL)</li> </ul>
<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 404 Not Found
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>profile</b>: UserProfileDto, zawiera: <ul style="list-style-type: none"> <li>– <b>username</b>: String</li> <li>– <b>profilePhoto</b>: String (URL)</li> <li>– <b>followersCount</b>: Integer</li> <li>– <b>followedCount</b>: Integer</li> <li>– <b>friendsCount</b>: Integer</li> <li>– <b>photosCount</b>: Integer</li> <li>– <b>mostPopularPhotos</b>: List&lt;ImageDto&gt;</li> </ul> </li> <li>• <b>friendStatus</b>: UserFriendStatus</li> <li>• <b>isFollowing</b>: Boolean</li> <li>• <b>isOwnProfile</b>: Boolean</li> </ul>

**Tabela 7.9:** Karta endpointu: /public/user-dashboard/profile/{targetUsername}

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP02
<b>Ścieżka:</b>	/user-dashboard/favorite-spots
<b>Nazwa:</b>	Pobierz listę ulubionych spotów użytkownika

<b>Parametry wejściowe:</b>	Brak
<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>type</b>: FavoriteSpotsListType (typ listy: ulubione, odwiedzone oraz do odwiedzenia)</li> <li>• <b>page</b>: Integer (opcjonalnie, domyślnie 0)</li> <li>• <b>size</b>: Integer (opcjonalnie, domyślnie 10)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 401 Unauthorized
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>items</b>: List&lt;FavoriteSpotDto&gt;, gdzie każdy element zawiera: <ul style="list-style-type: none"> <li>– <b>id</b>: Long</li> <li>– <b>name</b>: String</li> <li>– <b>country</b>: String</li> <li>– <b>city</b>: String</li> <li>– <b>description</b>: String</li> <li>– <b>rating</b>: Double</li> <li>– <b>viewsCount</b>: Integer</li> <li>– <b>imageUrl</b>: String</li> <li>– <b>type</b>: FavoriteSpotsListType</li> <li>– <b>coords</b>: SpotCoordinatesDto</li> <li>– <b>tags</b>: Set&lt;SpotTagDto&gt;</li> </ul> </li> <li>• <b>hasNext</b>: boolean</li> </ul>

**Tabela 7.10:** Karta endpointu: /user-dashboard/favorite-spots

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP03
<b>Ścieżka:</b>	/user-dashboard/photos
<b>Nazwa:</b>	Pobierz posortowane zdjęcia użytkownika
<b>Parametry wejściowe:</b>	Brak
<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>type:</b> DateSortType (typ sortowania po dacie)</li> <li>• <b>from:</b> LocalDate (opcjonalnie, data od)</li> <li>• <b>to:</b> LocalDate (opcjonalnie, data do)</li> <li>• <b>page:</b> Integer (opcjonalnie, domyślnie 0)</li> <li>• <b>size:</b> Integer (opcjonalnie, domyślnie 20)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 400 Bad Request, 401 Unauthorized
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>items:</b> List&lt;DatedMediaGroupDto&gt;, gdzie:           <ul style="list-style-type: none"> <li>– <b>date:</b> LocalDate (data grupy)</li> <li>– <b>media:</b> List&lt;MediaDto&gt;, każdy element:               <ul style="list-style-type: none"> <li>* <b>src:</b> String (URL)</li> <li>* <b>heartsCount:</b> Integer</li> <li>* <b>viewsCount:</b> Integer</li> <li>* <b>id:</b> Long</li> </ul> </li> </ul> </li> <li>• <b>hasNext:</b> boolean</li> </ul>

**Tabela 7.11:** Karta endpointu: /user-dashboard/photos

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP04

<b>Ścieżka:</b>	/user-dashboard/add-spot
<b>Nazwa:</b>	Pobierz listę spotów dodanych przez użytkownika
<b>Parametry wejściowe:</b>	Brak
<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>page</b>: Integer (opcjonalnie, domyślnie 0)</li> <li>• <b>size</b>: Integer (opcjonalnie, domyślnie 20)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 401 Unauthorized
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>items</b>: List&lt;AddSpotDto&gt;, każdy element: <ul style="list-style-type: none"> <li>– <b>id</b>: Long</li> <li>– <b>name</b>: String</li> <li>– <b>description</b>: String</li> <li>– <b>country</b>: String</li> <li>– <b>region</b>: String</li> <li>– <b>city</b>: String</li> <li>– <b>street</b>: String</li> <li>– <b>borderPoints</b>: List&lt;BorderPoint&gt; (x, y)</li> <li>– <b>firstPhotoUrl</b>: String</li> </ul> </li> <li>• <b>hasNext</b>: boolean</li> </ul>

**Tabela 7.12:** Karta endpointu: /user-dashboard/add-spot

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP05

<b>Ścieżka:</b>	/user-dashboard/add-spot
<b>Nazwa:</b>	Dodaj nowy spot użytkownika
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li><b>spot:</b> String (część multipart, JSON z danymi nowego spotu)</li> <li><b>media:</b> List&lt;MultipartFile&gt; (część multipart, pliki multimedialne spota)</li> </ul>
<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 404 Not Found, 401 Unauthorized
<b>Dane zwracane:</b>	Brak (pusta odpowiedź)

**Tabela 7.13:** Karta endpointu: /user-dashboard/add-spot

### Spoty i pogoda

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP06
<b>Ścieżka:</b>	/public/spot/gallery
<b>Nazwa:</b>	Pobierz stronę galerii mediów dla spota
<b>Parametry wejściowe:</b>	Brak

<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>spotId</b>: Long (identyfikator spota)</li> <li>• <b>mediaType</b>: String (typ plików, wartość enum GenericMediaType, PHOTO, VIDEO)</li> <li>• <b>sorting</b>: String (kryterium sortowania, po dacie / popularności)</li> <li>• <b>page</b>: Integer (numer strony, domyślnie 0)</li> <li>• <b>size</b>: Integer (rozmiar strony, domyślnie 6)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 404 Not Found
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>Page&lt;SpotSidebarMediaGalleryDto&gt;</b>: stronicowana lista elementów galerii, gdzie każdy element zawiera: <ul style="list-style-type: none"> <li>– <b>id</b>: Long (identyfikator media)</li> <li>– <b>url</b>: String (URL pliku)</li> <li>– <b>mediaType</b>: GenericMediaType (typ pliku)</li> </ul> </li> </ul>

**Tabela 7.14:** Karta endpointu: /public/spot/gallery

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP07
<b>Ścieżka:</b>	/public/spot/current-view
<b>Nazwa:</b>	Pobierz listę spotów w aktualnym widoku mapy
<b>Parametry wejściowe:</b>	Brak

<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>swLng</b>: double (długość geograficzna lewego dolnego rogu)</li> <li>• <b>swLat</b>: double (szerokość geograficzna lewego dolnego rogu)</li> <li>• <b>neLng</b>: double (długość geograficzna prawego górnego rogu)</li> <li>• <b>neLat</b>: double (szerokość geograficzna prawego górnego rogu)</li> <li>• <b>name</b>: String (fragment nazwy spotu, domyślnie pusty)</li> <li>• <b>sorting</b>: String (tryb sortowania, domyślnie <b>none</b>)</li> <li>• <b>ratingFrom</b>: double (minimalna ocena, domyślnie 0.0)</li> <li>• <b>page</b>: Integer (numer strony, domyślnie 0)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>Page&lt;SearchSpotDto&gt;</b>: stronicowana lista spotów, gdzie każdy element zawiera: <ul style="list-style-type: none"> <li>– <b>id</b>: Long (identyfikator spota)</li> <li>– <b>name</b>: String (nazwa spota)</li> <li>– <b>rating</b>: Double (0–5)</li> <li>– <b>ratingCount</b>: Integer (liczba ocen)</li> <li>– <b>firstPhoto</b>: String (URL pierwszego zdjęcia)</li> <li>– <b>tags</b>: Set&lt;SpotTagDto&gt; (tagi spota)</li> <li>– <b>centerPoint</b>: BorderPoint (środek obszaru spota)</li> </ul> </li> </ul>

**Tabela 7.15:** Karta endpointu: /public/spot/current-view

### KARTA ENDPOINTU API

<b>Identyfikator:</b>	EP08
<b>Ścieżka:</b>	/public/spot/get-spot-basic-weather
<b>Nazwa:</b>	Pobierz podstawowe informacje pogodowe dla wskazanej lokalizacji
<b>Parametry wejściowe:</b>	Brak
<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>latitude</b>: double (szerokość geograficzna)</li> <li>• <b>longitude</b>: double (długość geograficzna)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>temperature</b>: Double (temperatura)</li> <li>• <b>weatherCode</b>: int (kod warunków pogodowych)</li> <li>• <b>windSpeed</b>: Double (prędkość wiatru)</li> <li>• <b>isDay</b>: boolean (czy jest dzień)</li> </ul>

**Tabela 7.16:** Karta endpointu: /public/spot/get-spot-basic-weather

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP09
<b>Ścieżka:</b>	/public/spot/get-spot-detailed-weather
<b>Nazwa:</b>	Pobierz szczegółowe informacje pogodowe dla wskazanej lokalizacji
<b>Parametry wejściowe:</b>	Brak
<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>latitude</b>: double (szerokość geograficzna)</li> <li>• <b>longitude</b>: double (długość geograficzna)</li> </ul>

<b>Kod(y) statusu odpowiedzi:</b>	200 OK
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>temperature:</b> Double (temperatura)</li> <li>• <b>weatherCode:</b> int (kod warunków pogodowych)</li> <li>• <b>precipitationProbability:</b> Double (prawdopodobieństwo opadów)</li> <li>• <b>dewPoint:</b> Double (punkt rosy)</li> <li>• <b>relativeHumidity:</b> Double (wilgotność względna)</li> <li>• <b>isDay:</b> boolean (czy jest dzień)</li> <li>• <b>uvIndexMax:</b> Double (maksymalny indeks UV)</li> </ul>

**Tabela 7.17:** Karta endpointu: /public/spot/get-spot-detailed-weather

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP10
<b>Ścieżka:</b>	/public/spot/get-spot-wind-speeds
<b>Nazwa:</b>	Pobierz prędkości wiatru dla spotu na różnych wysokościach
<b>Parametry wejściowe:</b>	Brak
<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>latitude:</b> double (szerokość geograficzna)</li> <li>• <b>longitude:</b> double (długość geograficzna)</li> <li>• <b>spotId:</b> long (identyfikator spota)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK

<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>windSpeeds100m</b>: Double (prędkość wiatru na wysokości 100 metrów)</li> <li>• <b>windSpeeds200m</b>: Double (prędkość wiatru na wysokości 200 metrów)</li> <li>• <b>windSpeeds300m</b>: Double (prędkość wiatru na wysokości 300 metrów)</li> <li>• <b>windSpeeds500m</b>: Double (prędkość wiatru na wysokości 500 metrów)</li> <li>• <b>windSpeeds750m</b>: Double (prędkość wiatru na wysokości 750 metrów)</li> <li>• <b>windSpeeds1000m</b>: Double (prędkość wiatru na wysokości 1000 metrów)</li> </ul>
-----------------------	--

**Tabela 7.18:** Karta endpointu: /public/spot/get-spot-wind-speeds

#### Wyszukiwarka spotów

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP11
<b>Ścieżka:</b>	/public/spot/most-popular
<b>Nazwa:</b>	Pobierz 18 najpopularniejszych spotów
<b>Parametry wejściowe:</b>	Brak
<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	200 OK

<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>items:</b> List&lt;TopRatedSpotDto&gt; każdy element zawiera:           <ul style="list-style-type: none"> <li>– <b>id:</b> Long (identyfikator spota)</li> <li>– <b>name:</b> String (nazwa spota)</li> <li>– <b>city:</b> String (miasto, w którym znajduje się spot)</li> <li>– <b>imageUrl:</b> String (URL zdjęcia spota)</li> </ul> </li> </ul>
-----------------------	---

**Tabela 7.19:** Karta endpointu: /public/spot/most-popular

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP12
<b>Ścieżka:</b>	/public/spot/search/home-page
<b>Nazwa:</b>	Wyszukaj spedy na stronie głównej na podstawie lokalizacji
<b>Parametry wejściowe:</b>	Brak
<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>country:</b> String (opcjonalnie, kraj)</li> <li>• <b>region:</b> String (opcjonalnie, region)</li> <li>• <b>city:</b> String (opcjonalnie, miasto)</li> <li>• <b>userLongitude:</b> Double (opcjonalnie, długość geograficzna użytkownika)</li> <li>• <b>userLatitude:</b> Double (opcjonalnie, szerokość geograficzna użytkownika)</li> <li>• <b>page:</b> Integer (numer strony, domyślnie 0)</li> <li>• <b>size:</b> Integer (rozmiar strony, domyślnie 20)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK

<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>items:</b> List&lt;HomePageSpotDto&gt; (lista znalezionych spotów), każdy element:           <ul style="list-style-type: none"> <li>– <b>id:</b> Long (identyfikator spota)</li> <li>– <b>name:</b> String (nazwa spota)</li> <li>– <b>rating:</b> Double (średnia ocena spota)</li> <li>– <b>ratingCount:</b> Integer (liczba ocen spota)</li> <li>– <b>firstPhoto:</b> String (URL pierwszego zdjęcia spota)</li> <li>– <b>tags:</b> Set&lt;SpotTagDto&gt; (zestaw tagów przypisanych do spota)</li> <li>– <b>centerPoint:</b> BorderPoint (punkt centralny obszaru spota)</li> <li>– <b>city:</b> String (miasto, w którym znajduje się spot)</li> <li>– <b>distanceToUser:</b> Double (odległość od lokalizacji użytkownika, jeśli dostępna)</li> </ul> </li> <li>• <b>hasNext:</b> boolean (czy istnieje kolejna strona wyników)</li> </ul>
-----------------------	--

**Tabela 7.20:** Karta endpointu: /public/spot/search/home-page

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP13
<b>Ścieżka:</b>	/public/spot/search/home-page/locations
<b>Nazwa:</b>	Pobierz listę podpowiedzi lokalizacji dla wyszukiwarki na stronie głównej
<b>Parametry wejściowe:</b>	Brak

<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>query</b>: String (frazą wyszukiwania)</li> <li>• <b>type</b>: String (typ lokalizacji, kraj/region/miasto)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>List&lt;String&gt;</b>: lista podpowiedzi (nazwy lokalizacji)</li> </ul>

**Tabela 7.21:** Karta endpointu: /public/spot/search/home-page/locations

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP14
<b>Ścieżka:</b>	/public/spot/search/home-page/advance
<b>Nazwa:</b>	Wyszukaj spoty na stronie głównej (zaawansowane filtrowanie)
<b>Parametry wejściowe:</b>	Brak
<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>city</b>: String (opcjonalnie, miasto wyszukiwania)</li> <li>• <b>tags</b>: List&lt;String&gt; (opcjonalnie, lista tagów spota)</li> <li>• <b>userLongitude</b>: Double (opcjonalnie, długość geograficzna użytkownika)</li> <li>• <b>userLatitude</b>: Double (opcjonalnie, szerokość geograficzna użytkownika)</li> <li>• <b>sort</b>: SpotSortType (opcjonalnie, typ sortowania wyników)</li> <li>• <b>filter</b>: SpotRatingFilterType (opcjonalnie, filtr po ocenie)</li> <li>• <b>page</b>: Integer (numer strony, domyślnie 0)</li> <li>• <b>size</b>: Integer (rozmiar strony, domyślnie 20)</li> </ul>

<b>Kod(y) statusu odpowiedzi:</b>	200 OK
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>items:</b> List&lt;HomePageSpotDto&gt; (lista znalezionych spotów), każdy element:             <ul style="list-style-type: none"> <li>– <b>id:</b> Long (identyfikator spota)</li> <li>– <b>name:</b> String (nazwa spota)</li> <li>– <b>rating:</b> Double (średnia ocena spota)</li> <li>– <b>ratingCount:</b> Integer (liczba ocen spota)</li> <li>– <b>firstPhoto:</b> String (URL pierwszego zdjęcia spota)</li> <li>– <b>tags:</b> Set&lt;SpotTagDto&gt; (zestaw tagów przypisanych do spota)</li> <li>– <b>centerPoint:</b> BorderPoint (punkt centralny obszaru spota)</li> <li>– <b>city:</b> String (miasto, w którym znajduje się spot)</li> <li>– <b>distanceToUser:</b> Double (odległość od lokalizacji użytkownika, jeśli dostępna)</li> </ul> </li> <li>• <b>hasNext:</b> boolean (czy istnieje kolejna strona wyników)</li> </ul>

**Tabela 7.22:** Karta endpointu: /public/spot/search/home-page/advance

#### Komentarze do spotów

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP15
<b>Ścieżka:</b>	/public/spot/{spotId}/comments
<b>Nazwa:</b>	Pobierz stronicowaną listę komentarzy dla wskazanego spota

<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>spotId</b>: Long (identyfikator spota w ścieżce URL)</li> </ul>
<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>page</b>: Integer (numer strony, domyślnie 0; rozmiar strony = 2)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 404 Not Found, 401 Unauthorized

<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>Page&lt;SpotCommentDto&gt;</b>: stronicowana lista komentarzy dla danego spota, każdy element zawiera:           <ul style="list-style-type: none"> <li>– <b>id</b>: Long (identyfikator komentarza)</li> <li>– <b>author</b>: SpotCommentAuthorDto (dane autora komentarza)</li> <li>– <b>text</b>: String (treść komentarza)</li> <li>– <b>rating</b>: Double (ocena spota wystawiona w komentarzu, 0–5)</li> <li>– <b>upvotes</b>: Integer (liczba głosów pozytywnych na komentarz)</li> <li>– <b>downvotes</b>: Integer (liczba głosów negatywnych na komentarz)</li> <li>– <b>publishDate</b>: LocalDateTime (data i godzina publikacji komentarza)</li> <li>– <b>isUpVoted</b>: Boolean (czy bieżący użytkownik oddał głos w górę na ten komentarz)</li> <li>– <b>isDownVoted</b>: Boolean (czy bieżący użytkownik oddał głos w dół na ten komentarz)</li> <li>– <b>numberOfMedia</b>: Integer (łączna liczba dołączonych plików multimedialnych)</li> <li>– <b>mediaList</b>: List&lt;SpotCommentMediaDto&gt; (lista pierwszych plików komentarza)</li> </ul> </li> </ul>
-----------------------	--

**Tabela 7.23:** Karta endpointu: /public/spot/{spotId}/comments

#### KARTA ENDPOINTU API

<b>Identyfikator:</b>	EP16
<b>Ścieżka:</b>	/public/spot/{spotId}/comments/{commentId}
<b>Nazwa:</b>	Pobierz pełną listę mediów powiązanych z komentarzem
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>spotId:</b> Long (identyfikator spota w ścieżce URL)</li> <li>• <b>commentId:</b> Long (identyfikator komentarza w ścieżce URL)</li> </ul>
<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 404 Not Found
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>List&lt;SpotCommentMediaDto&gt;:</b> pełna lista mediów powiązanych z komentarzem, każdy element:             <ul style="list-style-type: none"> <li>– <b>id:</b> Long (identyfikator pliku multimedialnego)</li> <li>– <b>url:</b> String (URL pliku, używany do pobrania/wyświetlenia)</li> <li>– <b>genericMediaType:</b> GenericMediaType (typ pliku, PHOTO lub VIDEO)</li> </ul> </li> </ul>

**Tabela 7.24:** Karta endpointu: /public/spot/{spotId}/comments/{commentId}

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP17
<b>Ścieżka:</b>	/spot/{spotId}/comments
<b>Nazwa:</b>	Dodaj nowy komentarz do wskazanego spota

<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>spotId</b>: Long (identyfikator spota w ścieżce URL)</li> <li>• <b>body</b>: SpotCommentAddDto (dane nowego komentarza), zawiera:           <ul style="list-style-type: none"> <li>– <b>text</b>: String (treść komentarza)</li> <li>– <b>rating</b>: Double (ocena spota w komentarzu, zakres 0–5)</li> <li>– <b>mediaFiles</b>: List&lt;MultipartFile&gt; (lista załączonych plików, zdjęcia/filmy)</li> </ul> </li> </ul>
<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	201 Created, 404 Not Found, 401 Unauthorized, 422 Unprocessable Entity
<b>Dane zwracane:</b>	Brak (pusta odpowiedź)

**Tabela 7.25:** Karta endpointu: /spot/{spotId}/comments

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP18
<b>Ścieżka:</b>	/spot/comments/{commentId}/vote
<b>Nazwa:</b>	Oddaj głos na komentarz (góra/dół)
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>commentId</b>: Long (identyfikator komentarza w ścieżce URL)</li> </ul>
<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>isUpvote</b>: boolean (true = głos w góre, false = głos w dół)</li> </ul>

<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 401 Unauthorized, 404 Not Found, 409 Conflict, 403 Forbidden
<b>Dane zwracane:</b>	Brak (pusta odpowiedź)

**Tabela 7.26:** Karta endpointu: /spot/comments/{commentId}/vote

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP19
<b>Ścieżka:</b>	/spot/comments/vote-type
<b>Nazwa:</b>	Pobierz informację, jak bieżący użytkownik zagłosował na komentarz
<b>Parametry wejściowe:</b>	Brak
<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>commentId:</b> Long (identyfikator komentarza)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 404 Not Found, 401 Unauthorized
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>voteInfo:</b> SpotCommentVoteType (typ oddanego głosu, UPVOTE, DOWNVOTE, NONE)</li> </ul>

**Tabela 7.27:** Karta endpointu: /spot/comments/vote-type

## Forum – posty

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP20
<b>Ścieżka:</b>	/public/post/{postId}
<b>Nazwa:</b>	Pobierz szczegółowe informacje o poście
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>postId:</b> Long (identyfikator posta w ścieżce URL)</li> </ul>
<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 404 Not Found

<p><b>Dane zwracane:</b></p>	<ul style="list-style-type: none"> <li>• <b>PostDetailsDto</b>, zawiera:</li> </ul> <ul style="list-style-type: none"> <li>– <b>id</b>: Long (identyfikator posta)</li> <li>– <b>title</b>: String (tytuł posta)</li> <li>– <b>content</b>: String (pełna treść posta)</li> <li>– <b>category</b>: ForumCategoryDto (kategoria forum, do której należy post)</li> <li>– <b>tags</b>: List&lt;ForumTagDto&gt; (lista tagów przypisanych do posta)</li> <li>– <b>author</b>: AuthorDto (dane autora posta)</li> <li>– <b>isAuthor</b>: Boolean (czy bieżący użytkownik jest autorem posta)</li> <li>– <b>isFollowed</b>: Boolean (czy bieżący użytkownik obserwuje ten post)</li> <li>– <b>publishDate</b>: LocalDateTime (data i godzina publikacji posta)</li> <li>– <b>views</b>: Integer (liczba wyświetleń posta)</li> <li>– <b>upVotes</b>: Integer (liczba głosów w górę na post)</li> <li>– <b>downVotes</b>: Integer (liczba głosów w dół na post)</li> <li>– <b>isUpVoted</b>: Boolean (czy bieżący użytkownik oddał głos w górę na post)</li> <li>– <b>isDownVoted</b>: Boolean (czy bieżący użytkownik oddał głos w dół na post)</li> <li>– <b>commentsCount</b>: Integer (łączna liczba komentarzy pod postem)</li> </ul>
------------------------------	--

**Tabela 7.28:** Karta endpointu: /public/post/{postId}

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP21
<b>Ścieżka:</b>	/post
<b>Nazwa:</b>	Dodaj nowy post na forum
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>body:</b> PostDto (dane nowego posta), zawiera:             <ul style="list-style-type: none"> <li>– <b>title:</b> String (tytuł posta)</li> <li>– <b>content:</b> String (pełna treść posta)</li> <li>– <b>category:</b> String (nazwa kategorii forum, do której ma trafić post)</li> <li>– <b>tags:</b> List&lt;String&gt; (lista nazw tagów przypisanych do posta)</li> </ul> </li> </ul>
<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	201 Created, 400 Bad Request, 404 Not Found, 409 Conflict, 422 Unprocessable Entity
<b>Dane zwracane:</b>	Brak (pusta odpowiedź)

**Tabela 7.29:** Karta endpointu: /post

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP22
<b>Ścieżka:</b>	/post/{postId}
<b>Nazwa:</b>	Usuń wybrany post
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>postId:</b> Long (identyfikator posta w ścieżce URL)</li> </ul>

<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	204 No Content, 401 Unauthorized, 404 Not Found
<b>Dane zwracane:</b>	Brak (pusta odpowiedź)

**Tabela 7.30:** Karta endpointu: /post/{postId}

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP23
<b>Ścieżka:</b>	/post/{postId}/vote
<b>Nazwa:</b>	Oddaj głos na post (góra/dół)
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>postId:</b> Long (identyfikator posta w ścieżce URL)</li> </ul>
<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>isUpvote:</b> boolean (true = głos w góre, false = głos w dół)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 401 Unauthorized, 404 Not Found
<b>Dane zwracane:</b>	Brak (pusta odpowiedź)

**Tabela 7.31:** Karta endpointu: /post/{postId}/vote

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP24
<b>Ścieżka:</b>	/public/categories-tags
<b>Nazwa:</b>	Pobierz listę wszystkich kategorii i tagów forum
<b>Parametry wejściowe:</b>	Brak
<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	200 OK
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>ForumCategoriesAndTagsDto</b> (zestaw kategorii i tagów forum), zawiera:           <ul style="list-style-type: none"> <li>– <b>categories</b>: List&lt;ForumCategoryDto&gt; (lista dostępnych kategorii), gdzie każdy element zawiera:               <ul style="list-style-type: none"> <li>* <b>id</b>: Long (identyfikator kategorii)</li> <li>* <b>name</b>: String (nazwa kategorii)</li> <li>* <b>description</b>: String (opis kategorii)</li> <li>* <b>colour</b>: String (kolor kategorii)</li> </ul> </li> <li>– <b>tags</b>: List&lt;ForumTagDto&gt; (lista dostępnych tagów), gdzie każdy element zawiera:               <ul style="list-style-type: none"> <li>* <b>id</b>: Long (identyfikator tagu)</li> <li>* <b>name</b>: String (nazwa tagu)</li> </ul> </li> </ul> </li> </ul>

**Tabela 7.32:** Karta endpointu: /public/categories-tags

#### Forum – komentarze do postów

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP25
<b>Ścieżka:</b>	/public/post/{postId}/comments
<b>Nazwa:</b>	Pobierz stronicowaną listę komentarzy posta
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>postId:</b> Long (identyfikator posta w ścieżce URL)</li> </ul>
<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>page:</b> Integer (numer strony, domyślnie 0)</li> <li>• <b>size:</b> Integer (liczba komentarzy na stronie, domyślnie 10)</li> <li>• <b>sortBy:</b> PostCommentSortField (pole sortowania, domyślnie PUBLISH_DATE)</li> <li>• <b>sortDirection:</b> SortDirection (kierunek sortowania, domyślnie DESC)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 404 Not Found

<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>Page&lt;PostCommentGeneralDto&gt;</b>, każdy element zawiera:           <ul style="list-style-type: none"> <li>– <b>id</b>: Long (identyfikator komentarza)</li> <li>– <b>content</b>: String (treść komentarza)</li> <li>– <b>upVotes</b>: Integer (liczba głosów w górę)</li> <li>– <b>downVotes</b>: Integer (liczba głosów w dół)</li> <li>– <b>repliesCount</b>: Integer (liczba odpowiedzi)</li> <li>– <b>publishDate</b>: LocalDateTime (data publikacji)</li> <li>– <b>author</b>: AuthorDto (dane autora)</li> <li>– <b>isAuthor</b>: Boolean (czy bieżący użytkownik jest autorem)</li> <li>– <b>isUpVoted</b>: Boolean (czy użytkownik zagłosował w górę)</li> <li>– <b>isDownVoted</b>: Boolean (czy użytkownik zagłosował w dół)</li> <li>– <b>isReply</b>: Boolean (czy komentarz jest odpowiedzią)</li> <li>– <b>isDeleted</b>: Boolean (czy komentarz został usunięty logicznie)</li> </ul> </li> </ul>
-----------------------	---

**Tabela 7.33:** Karta endpointu: /public/post/{postId}/comments

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP26
<b>Ścieżka:</b>	/post/{postId}/comments
<b>Nazwa:</b>	Dodaj nowy komentarz do posta

<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>postId</b>: Long (identyfikator posta w ścieżce URL)</li> <li>• <b>body</b>: PostCommentDto, zawiera:           <ul style="list-style-type: none"> <li>– <b>content</b>: String (treść komentarza)</li> </ul> </li> </ul>
<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	201 Created, 400 Bad Request, 404 Not Found, 422 Unprocessable Entity
<b>Dane zwracane:</b>	Brak (pusta odpowiedź)

**Tabela 7.34:** Karta endpointu: /post/{postId}/comments

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP27
<b>Ścieżka:</b>	/post/comments/{commentId}
<b>Nazwa:</b>	Edytuj istniejący komentarz do posta
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>commentId</b>: Long (identyfikator komentarza w ścieżce URL)</li> <li>• <b>body</b>: PostCommentDto, zawiera:           <ul style="list-style-type: none"> <li>– <b>content</b>: String (treść komentarza)</li> </ul> </li> </ul>
<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 422 Unprocessable Entity

<b>Dane zwracane:</b>	Brak (pusta odpowiedź)
-----------------------	------------------------

**Tabela 7.35:** Karta endpointu: /post/comments/{commentId}

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP28
<b>Ścieżka:</b>	/post/comments/{commentId}/vote
<b>Nazwa:</b>	Oddaj głos na komentarz (góra/dół)
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>commentId:</b> Long (identyfikator komentarza w ścieżce URL)</li> </ul>
<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>isUpvote:</b> boolean (true = głos w góre, false = głos w dół)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 403 Forbidden, 404 Not Found
<b>Dane zwracane:</b>	Brak (pusta odpowiedź)

**Tabela 7.36:** Karta endpointu: /post/comments/{commentId}/vote

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP29
<b>Ścieżka:</b>	/comments/{commentId}/replies
<b>Nazwa:</b>	Dodaj odpowiedź na komentarz

<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>commentId</b>: Long (identyfikator komentarza nadzędnego w ścieżce URL)</li> <li>• <b>body</b>: PostCommentDto, zawiera:           <ul style="list-style-type: none"> <li>– <b>content</b>: String (treść komentarza)</li> </ul> </li> </ul>
<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	201 Created, 400 Bad Request, 404 Not Found, 409 Conflict, 422 Unprocessable Entity
<b>Dane zwracane:</b>	Brak (pusta odpowiedź)

**Tabela 7.37:** Karta endpointu: /comments/{commentId}/replies

#### Konto użytkownika – rejestracja, logowanie, hasło

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP30
<b>Ścieżka:</b>	/public/account/register
<b>Nazwa:</b>	Zarejestruj nowego użytkownika
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>body</b>: UserRegisterDto, zawiera:           <ul style="list-style-type: none"> <li>– <b>username</b>: String</li> <li>– <b>email</b>: String</li> <li>– <b>password</b>: String</li> </ul> </li> </ul>
<b>Query params:</b>	Brak

<b>Kod(y) statusu odpowiedzi:</b>	201 Created, 401 Unauthorized, 409 Conflict, 422 Unprocessable Entity, 500 Internal Server Error
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>body:</b> String (komunikat)</li> <li>• JWT tokeny ustawione w ciasteczkach HTTP-only</li> </ul>

**Tabela 7.38:** Karta endpointu: /public/account/register

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP31
<b>Ścieżka:</b>	/public/account/login
<b>Nazwa:</b>	Zaloguj użytkownika
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>body:</b> UserLoginDto, zawiera:           <ul style="list-style-type: none"> <li>– <b>username:</b> String</li> <li>– <b>password:</b> String</li> </ul> </li> </ul>
<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 401 Unauthorized, 422 Unprocessable Entity
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• Brak (pusta odpowiedź w body)</li> <li>• JWT tokeny zwrócone w ciasteczkach HTTP-only</li> </ul>

**Tabela 7.39:** Karta endpointu: /public/account/login

KARTA ENDPOINTU API	
---------------------	--

<b>Identyfikator:</b>	EP32
<b>Ścieżka:</b>	/public/account/forgot-password
<b>Nazwa:</b>	Rozpocznij procedurę resetu hasła (wyślij link na e-mail)
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>body:</b> String (adres e-mail użytkownika w treści żądania)</li> </ul>
<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 404 Not Found, 422 Unprocessable Entity, 500 Internal Server Error
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>body:</b> String (komunikat)</li> <li>• Link resetujący hasło wysłany na podany adres e-mail</li> </ul>

**Tabela 7.40:** Karta endpointu: /public/account/forgot-password

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP33
<b>Ścieżka:</b>	/public/account/set-new-password
<b>Nazwa:</b>	Ustaw nowe hasło użytkownika na podstawie tokenu resetującego
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>body:</b> UserPasswordResetDto, zawiera:           <ul style="list-style-type: none"> <li>– <b>token:</b> String (UUID – token resetu hasła)</li> <li>– <b>password:</b> String</li> </ul> </li> </ul>
<b>Query params:</b>	Brak

<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 400 Bad Request, 404 Not Found, 422 Unprocessable Entity
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>body:</b> String (komunikat)</li> </ul>

**Tabela 7.41:** Karta endpointu: /public/account/set-new-password

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP34
<b>Ścieżka:</b>	/account/check
<b>Nazwa:</b>	Sprawdź, czy użytkownik jest uwierzytelniony
<b>Parametry wejściowe:</b>	Brak
<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 401 Unauthorized, 403 Forbidden
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• Brak (pusta odpowiedź; sam status informuje o uwierzytelnieniu)</li> </ul>

**Tabela 7.42:** Karta endpointu: /account/check

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP35

<b>Ścieżka:</b>	/account/login-success
<b>Nazwa:</b>	Obsłuż użytkownika zalogowanego przez OAuth2 i przekieruj go do aplikacji
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>Brak klasycznego body – endpoint wywoływany jest jako redirect callback po poprawnym logowaniu przez dostawcę OAuth2.</li> <li>Kontekst użytkownika przekazywany jest w obiekcie <code>OAuth2AuthenticationToken</code>.</li> </ul>
<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	302 Found (redirect), 404 Not Found, 409 Conflict, 422 Unprocessable Entity, 500 Internal Server Error
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>Przekierowanie użytkownika na stronę frontendową skonfigurowaną w <code>UrlsProperties.afterLoginPageUrl</code>.</li> </ul>

**Tabela 7.43:** Karta endpointu: /account/login-success

#### GIF-y (Tenor) – integracja czatu

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP36
<b>Ścieżka:</b>	/gifs/trending
<b>Nazwa:</b>	Pobierz listę trendujących kategorii GIF-ów z Tenor
<b>Parametry wejściowe:</b>	Brak
<b>Query params:</b>	Brak

<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 500 Internal Server Error
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>items</b>: List&lt;TenorGifCategoryDto&gt;, każdy element zawiera:             <ul style="list-style-type: none"> <li>– <b>searchTerm</b>: String (frazą wyszukiwania powiązana z kategorią)</li> <li>– <b>path</b>: String (ścieżka kategorii w Tenor)</li> <li>– <b>gifUrl</b>: String (URL GIF-a)</li> </ul> </li> </ul>

**Tabela 7.44:** Karta endpointu: /gifs/trending

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP37
<b>Ścieżka:</b>	/gifs/search
<b>Nazwa:</b>	Wyszukaj GIF-y po frazie tekstowej
<b>Parametry wejściowe:</b>	Brak
<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>searchPhrase</b>: String (frazą wyszukiwania)</li> <li>• <b>next</b>: String (token paginacji zwrócony z poprzedniego wywołania; dla pierwszego zapytania może być pusty)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 500 Internal Server Error

<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>body:</b> TenorGifSearchWrapperDto (wyniki wyszukiwania GIF-ów), zawiera:           <ul style="list-style-type: none"> <li>– <b>gifs:</b> List&lt;TenorGifSearchDto&gt; (lista pasujacych GIF-ów), każdy element:               <ul style="list-style-type: none"> <li>* <b>url:</b> String (URL GIF-a)</li> </ul> </li> <li>– <b>next:</b> String (token do pobrania kolejnej strony wyników)</li> </ul> </li> </ul>
-----------------------	---

**Tabela 7.45:** Karta endpointu: /gifs/search

## Czat – REST API

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP38
<b>Ścieżka:</b>	/chats/{chatId}/messages
<b>Nazwa:</b>	Pobierz stronicowane wiadomości dla wybranego czatu
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>chatId:</b> Long (identyfikator czatu)</li> </ul>
<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>pageParam:</b> Integer (numer strony wiadomości, domyślnie 1 – pierwsza strona po wstępnym pobraniu)</li> <li>• <b>numberOfMessagesPerPage:</b> Integer (liczba wiadomości na stronę, domyślnie 20)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 401 Unauthorized, 404 Not Found

<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>body:</b> ChatMessageDtoSlice, zawiera:           <ul style="list-style-type: none"> <li>– <b>messages:</b> List&lt;ChatMessageDto&gt;, każdy element:               <ul style="list-style-type: none"> <li>* <b>id:</b> Long (identyfikator wiadomości)</li> <li>* <b>sender:</b> ChatMessageSenderDto (dane nadawcy wiadomości)</li> <li>* <b>sentAt:</b> LocalDateTime (data i godzina wysłania wiadomości)</li> <li>* <b>content:</b> String (treść wiadomości; dla wiadomości plikowych może być pusty)</li> <li>* <b>chatId:</b> Long (identyfikator czatu, do którego należy wiadomość)</li> <li>* <b>attachedFiles:</b> List&lt;ChatMessageAttachedFileDto&gt; (lista załączonych plików)</li> </ul> </li> <li>– <b>hasNextSlice:</b> Boolean (czy istnieje kolejna „strona” / porcja wiadomości)</li> <li>– <b>numberOfMessages:</b> Integer (liczba wiadomości zwróconych w tej odpowiedzi)</li> <li>– <b>sliceNumber:</b> Integer (numer bieżącej porcji wiadomości)</li> </ul> </li> </ul>
-----------------------	--

**Tabela 7.46:** Karta endpointu: /chats/{chatId}/messages

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP39
<b>Ścieżka:</b>	/chats/get-or-create-private-chat

<b>Nazwa:</b>	Pobierz istniejący lub utwórz nowy prywatny czat z użytkownikiem
<b>Parametry wejściowe:</b>	Brak
<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>receiverUsername:</b> String (nazwa użytkownika, z którym chcemy rozpocząć lub kontynuować rozmowę)</li> <li>• <b>chatId:</b> Long (opcjonalnie, identyfikator istniejącego czatu – jeśli jest już znany)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 401 Unauthorized, 404 Not Found
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>body:</b> ChatDto (szczegóły czatu), zawiera: <ul style="list-style-type: none"> <li>– <b>id:</b> Long (identyfikator czatu)</li> <li>– <b>name:</b> String (nazwa czatu – nazwa grupy lub nazwa rozmówcy)</li> <li>– <b>lastMessage:</b> ChatMessageDto (ostatnia wiadomość w czacie, jeśli istnieje)</li> <li>– <b>imageUrl:</b> String (URL avatara czatu lub rozmówcy)</li> <li>– <b>messages:</b> List&lt;ChatMessageDto&gt; (lista wiadomości zwróconych razem z czatem)</li> <li>– <b>chatType:</b> ChatType (typ czatu: PRIVATE lub GROUP)</li> <li>– <b>participants:</b> List&lt;ChatParticipantDto&gt; (lista uczestników czatu)</li> </ul> </li> </ul>

**Tabela 7.47:** Karta endpointu: /chats/get-or-create-private-chat

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP40
<b>Ścieżka:</b>	/chats/{chatId}/send-files
<b>Nazwa:</b>	Wyślij jeden lub wiele plików w ramach czatu
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>chatId:</b> Long (identyfikator czatu w ścieżce)</li> <li>• <b>media:</b> List&lt;MultipartFile&gt; (lista załączanych plików do wysłania w wiadomości)</li> </ul>
<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	201 Created, 401 Unauthorized, 404 Not Found, 413 Payload Too Large, 415 Unsupported Media Type, 500 Internal Server Error
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• Brak (pusta odpowiedź; wiadomości z plikami pojawią się w historii czatu)</li> </ul>

**Tabela 7.48:** Karta endpointu: /chats/{chatId}/send-files

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP41
<b>Ścieżka:</b>	/chats/create/group
<b>Nazwa:</b>	Utwórz nowy czat grupowy
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>body:</b> CreateGroupChatDto (dane nowego czatu grupowego), zawiera:           <ul style="list-style-type: none"> <li>– <b>usermes:</b> List&lt;String&gt; (lista nazw użytkowników, którzy mają zostać uczestnikami czatu)</li> <li>– <b>ownerUsername:</b> String (nazwa właściciela / twórcy czatu)</li> </ul> </li> </ul>

<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	201 Created, 400 Bad Request, 401 Unauthorized
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>body:</b> ChatDto (utworzony czat grupowy), zawiera:             <ul style="list-style-type: none"> <li>– <b>id:</b> Long (identyfikator czatu)</li> <li>– <b>name:</b> String (nazwa czatu – nazwa grupy)</li> <li>– <b>lastMessage:</b> ChatMessageDto (ostatnia wiadomość w czacie, jeśli istnieje)</li> <li>– <b>imageUrl:</b> String (URL avatara czatu)</li> <li>– <b>messages:</b> List&lt;ChatMessageDto&gt; (lista wiadomości zwróconych razem z czatem)</li> <li>– <b>chatType:</b> ChatType (typ czatu, PRIVATE lub GROUP)</li> <li>– <b>participants:</b> List&lt;ChatParticipantDto&gt; (lista uczestników czatu)</li> </ul> </li> </ul>

**Tabela 7.49:** Karta endpointu: /chats/create/group

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP42
<b>Ścieżka:</b>	/chats/{chatId}
<b>Nazwa:</b>	Zaktualizuj dane czatu grupowego (nazwa, zdjęcie)

<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>chatId</b>: Long (identyfikator czatu grupowego)</li> <li>• <b>updateGroupChatDto</b>: UpdateGroupChatDto (wysyłany jako multipart/form-data, zawiera dane do zmiany, nowa nazwa, nowe zdjęcie)</li> </ul>
<b>Query params:</b>	Brak
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 401 Unauthorized, 404 Not Found, 413 Payload Too Large, 415 Unsupported Media Type, 500 Internal Server Error
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>body</b>: UpdatedGroupChatDto (zaktualizowane dane czatu grupowego), zawiera: <ul style="list-style-type: none"> <li>– <b>newName</b>: String (aktualna nazwa czatu po zmianie)</li> <li>– <b>newImgUrl</b>: String (aktualny URL obrazka grupy po zmianie)</li> </ul> </li> </ul>

**Tabela 7.50:** Karta endpointu: /chats/{chatId}

KARTA ENDPOINTU API	
<b>Identyfikator:</b>	EP43
<b>Ścieżka:</b>	/chats/group-chat/add/search/{chatId}
<b>Nazwa:</b>	Wyszukaj potencjalnych użytkowników do dodania do czatu grupowego
<b>Parametry wejściowe:</b>	<ul style="list-style-type: none"> <li>• <b>chatId</b>: Long (identyfikator czatu grupowego, do którego chcemy dodać użytkowników)</li> </ul>

<b>Query params:</b>	<ul style="list-style-type: none"> <li>• <b>query</b>: String (fraza wyszukiwania po nazwie użytkownika)</li> <li>• <b>page</b>: Integer (numer strony wyników, domyślnie 0)</li> <li>• <b>size</b>: Integer (liczba wyników na stronę, domyślnie 20)</li> </ul>
<b>Kod(y) statusu odpowiedzi:</b>	200 OK, 401 Unauthorized, 404 Not Found
<b>Dane zwracane:</b>	<ul style="list-style-type: none"> <li>• <b>body</b>: SimpleSliceDto&lt;PotentialChatMemberDto&gt; (stworzona lista potencjalnych uczestników czatu), zawiera: <ul style="list-style-type: none"> <li>– <b>hasNext</b>: boolean (czy istnieje kolejna „strona” wyników)</li> <li>– <b>collection</b>: Collection&lt;PotentialChatMemberDto&gt; (kolekcja potencjalnych użytkowników), każdy element: <ul style="list-style-type: none"> <li>* <b>username</b>: String (nazwa użytkownika)</li> <li>* <b>profileImg</b>: String (URL zdjęcia profilowego użytkownika)</li> </ul> </li> </ul> </li> </ul>

**Tabela 7.51:** Karta endpointu: /chats/group-chat/add/search/{chatId}

### 7.1.3 Integracja z bazą danych

W aplikacji wykorzystano relacyjną **bazę danych** PostgreSQL, która w środowisku deweloperskim uruchamiana jest jako kontener w aplikacji Docker. Komunikacja **backendu** z bazą danych odbywa się z wykorzystaniem wzorca Repository oraz **bibliotek** oferowanych przez Spring Boot, co umożliwia efektywne zarządzanie danymi oraz utrzymanie spójności warstwy dostępu do danych.

W systemie zaimplementowano zestaw najistotniejszych tabel, które opisano poniżej:

- **chat-invitations** — przechowuje zaproszenia do czatów wysyłane użytkownikom.
- **chat-message-attached-file** — przechowuje pliki dołączone do wiadomości w czatach.
- **chat-messages** — zapisuje wiadomości wysyłane w czatach.
- **chat-participants** — zawiera informacje o uczestnikach poszczególnych czatów.
- **chats** — lista czatów dostępnych w systemie.
- **favorite-spots** — informacje o miejscach (spotach) oznaczonych jako ulubione przez użytkowników.
- **forum-categories** — kategorie, do których przypisywane są posty na forum.
- **forum-tags** — tagi przypisywane postom na forum.
- **friendships** — relacje znajomości między użytkownikami.
- **media** — ogólne media przesyłane przez użytkowników na forum (zdjęcia, filmy).
- **post-comment-down-votes** — przechowuje „minusy” nadawane komentarzom do postów.
- **post-comment-reports** — raporty zgłasiane przez użytkowników wobec komentarzy.
- **post-comment-up-votes** — przechowuje „plusy” nadawane komentarzom do postów.
- **post-comments** — komentarze użytkowników do postów.
- **post-down-votes** — „minusy” nadawane postom.
- **post-followers** — informacje o użytkownikach obserwujących dany post.

- **post-reports** — raporty zgłasiane wobec postów.
- **post-tags** — tagi przypisane do konkretnych postów.
- **post-up-votes** — „plusy” nadawane postom.
- **posts** — posty tworzone przez użytkowników.
- **spot-comment-down-votes** — „minusy” nadawane komentarzom do spotów.
- **spot-comment-media** — pliki multimedialne dołączone do komentarzy przy spotach.
- **spot-comment-up-votes** — „plusy” nadawane komentarzom do spotów.
- **spot-comments** — komentarze użytkowników do spotów.
- **spot-media** — pliki multimedialne związane z konkretnymi spotami.
- **spots** — baza spotów w systemie.
- **spots-tags** — tagi przypisane do poszczególnych spotów.
- **tags-of-spots** — alternatywna tabela z tagami dla spotów.
- **user-followed-posts** — lista postów śledzonych przez użytkowników.
- **user-followers** — relacje obserwujących użytkowników.
- **user-liked-spot-media** — informacja o polubieniach mediów powiązanych ze spotami.
- **users** — dane użytkowników systemu.

#### **7.1.4 Obsługa uwierzytelnienia**

#### **7.1.5 Konteneryzacja**

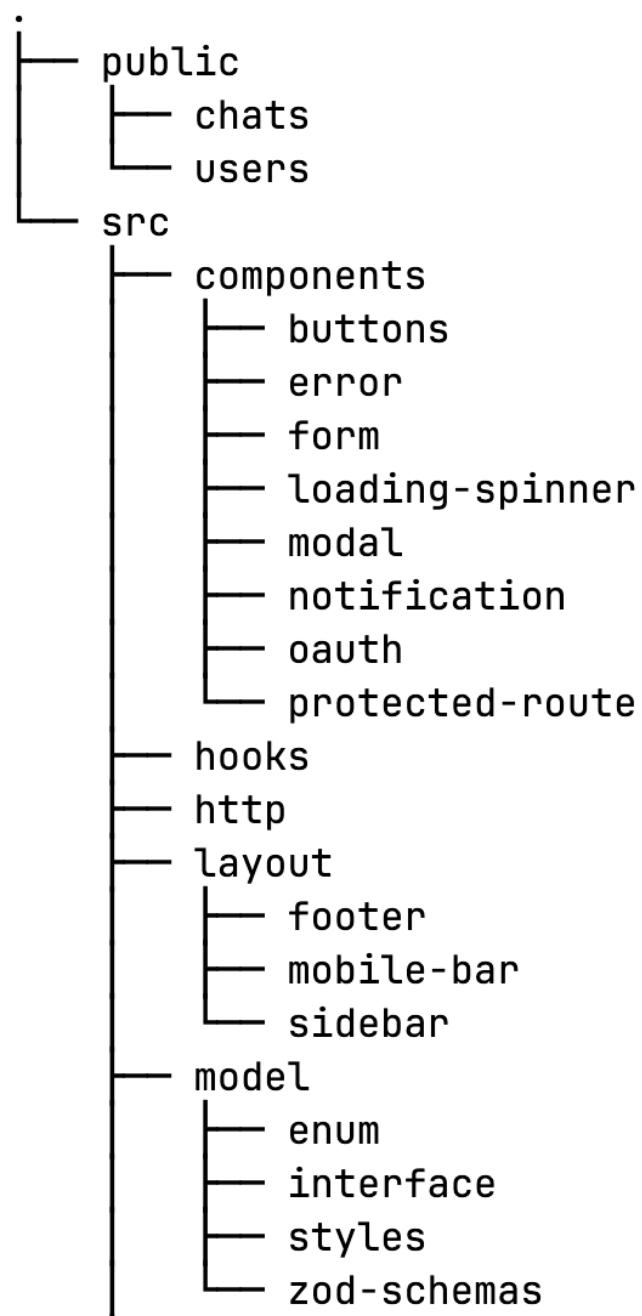
### **7.2 Implementacja frontendu**

W niniejszym rozdziale przedstawiono proces implementacji części [frontendowej](#) aplikacji.

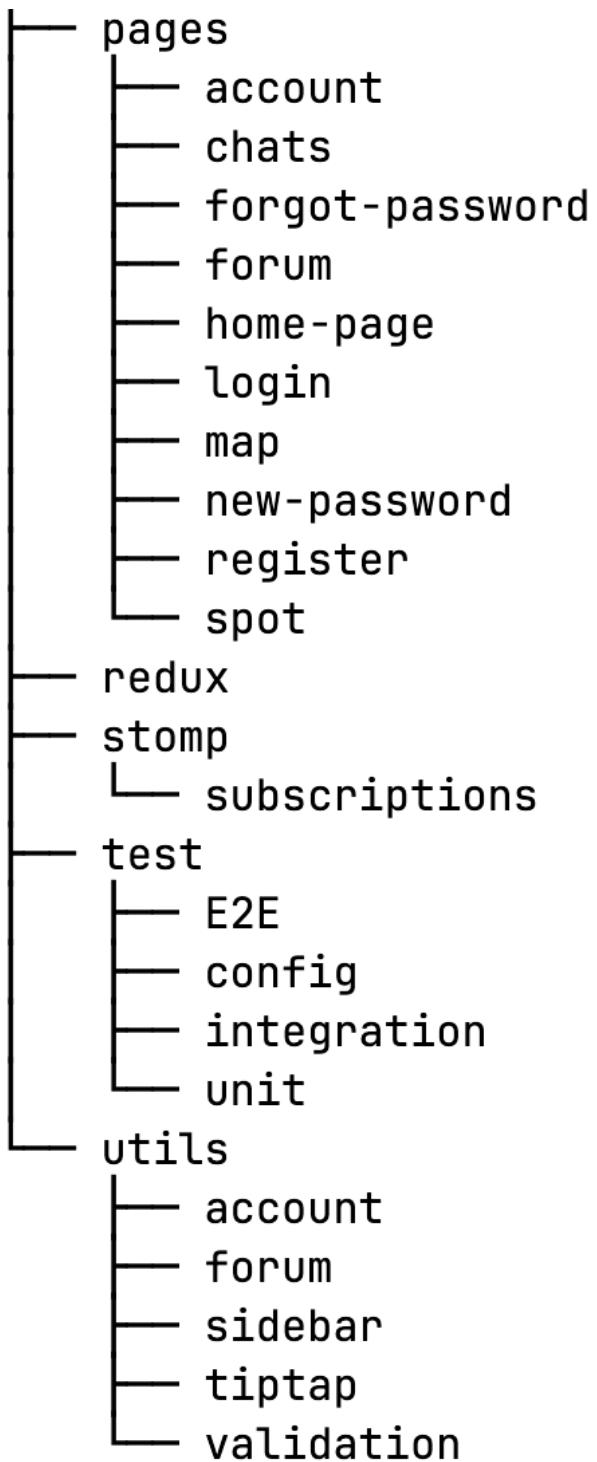
#### **7.2.1 Struktura aplikacji**

W niniejszym podrozdziale przedstawiona została struktura aplikacji [frontendowej](#) oraz organizację jej kluczowych elementów.

Architekturę aplikacji [frontendowej](#) zaprojektowano w strukturze [Folder by type](#), która polega na podziale kodu według typu zasobu (komponenty, strony, modele itd.). Każdy plik znajduje się w katalogu odpowiadającym jego przeznaczeniu, co przedstawiono na rysunkach [7.3](#) oraz [7.4](#).



Rysunek 7.3: Struktura katalogów (1)



Rysunek 7.4: Struktura katalogów (2)

Głównym elementem aplikacji jest mechanizm routingu oparty na [Bibliotece React Router](#). Definiuje on ścieżki do poszczególnych funkcjonalności aplikacji. Dzięki temu możliwa jest płynna nawigacja między różnymi widokami bez konieczności przeładowywania strony.

```
const router : Router = createBrowserRouter([
  {
    path: "/",
    element: <Layout />,
    errorElement: <Error error={undefined} />,
    children: [
      {
        index: true,
        element: <HomePage />,
      },
      {
        path: "advanced",
        element: <AdvanceHomePage />,
      },
      {
        path: "account",
        children: [ 11 elements... ],
      },
      {
        path: "register",
        element: <Register />,
      },
      {
        path: "login",
        element: <Login />,
      },
      {
        path: "forgot-password",
        element: <ForgotPassword />,
      },
    ],
  },
])
```

Rysunek 7.5: Implementacja routera (1)

```
{  
  path: "new-password",  
  element: <NewPassword />,  
},  
{  
  path: "forum",  
  element: <Forum />,  
},  
{  
  path: "forum/:postId/:slugTitle?",  
  element: <ForumThread />,  
},  
{  
  path: "map",  
  element: <MapPage />,  
},  
{  
  path: "chat",  
  element: (  
    <ProtectedRoute>  
      <ChatsPage />  
    </ProtectedRoute>  
  ),  
},  
],  
);  
  
export default router; Show usages ⚡ Adam Langmesser
```

Rysunek 7.6: Implementacja routera (2)

W projekcie zastosowano również wzorzec **protected route**, który służy do zabezpieczania wybranych tras przed dostępem użytkowników niezalogowanych. W pliku **router.tsx**, znajdującym się w głównym katalogu projektu, w konfiguracji przekazywanej do funkcji **createBrowserRouter** (rysunki 7.5 oraz 7.6), wybrane ścieżki opakowano w komponent **ProtectedRoute**. Komponent ten pełni

rolę bramki (rysunek 7.7).

Przykładem takiej chronionej ścieżki jest trasa `/chat`, prowadząca do modułu czatu dostępnego wyłącznie dla zalogowanych użytkowników. Jeśli niezalogowany użytkownik spróbuje uzyskać dostęp do tej ścieżki, zostanie automatycznie przekierowany na stronę główną.

```
export default function ProtectedRoute({ children }) { Show usages & Mredosz
  const isLoggedIn = useSelector(state) => state.account.isLoggedIn;

  return isLoggedIn ? children : <Navigate to="/" />;
}
```

Rysunek 7.7: Implementacja komponentu bramki (`ProtectedRoute`)

### 7.2.2 Zarządzanie stanem i przepływ danych

W niniejszym podrozdziale opisano zastosowane w projekcie podejście do zarządzania **stanem** oraz organizację przepływu danych w aplikacji frontendowej.

W projekcie postawiono na zrównoważone podejście do zarządzania **stanem**. Korzysta się zarówno z lokalnego **stanu** komponentów (za pomocą **hooka useState**) [14], jak i ze **stanu** globalnego, utrzymywanej przez bibliotekę **React Redux** [15]. Globalny **stan** wprowadzono w celu możliwie jak największego ograniczenia przekazywanie **propsów** w głąb drzewa komponentów oraz uniknąć niepotrzebnych ponownych renderów.

Do przechowywania **stanu** lokalnego, ograniczonego tylko do danego komponentu (lub jego najbliższych elementów podlegających), wykorzystuje się **hook useState**. Natomiast efekty uboczne i synchronizację realizuje się za pomocą **useEffect**. W przypadku bardziej złożonej logiki lub potrzeby ponownego wykorzystania kodu powstały **hooki** niestandardowe, takie jak **useScreenSize**, **useDarkMode** czy **useClickOutside**. Dzięki temu większość logiki prezentacji wydzielono z warstwy **UI**, co poprawia czytelność i ułatwia utrzymanie kodu.

Z racji tego, że korzystamy z **reacta** w połączeniu z **TypeScriptem**, przygoto-

wano również własne **hooki** wspomagające typowanie, takie jak `useDispatchTyped` oraz `useSelectorTyped`. Pozwalają one na bezpieczne typowanie akcji oraz selektorów **reduxa** bez konieczności powtarzania adnotacji typów w każdym komponencie. Fragmenty tej implementacji przedstawiono na rysunkach 7.8 oraz 7.9.

```

const store : EnhancedStore<{ account: AccountSliceProp... }> = configureStore({
  reducer: {
    account: accountSlice.reducer,
    notification: notificationSlice.reducer,
    spotDetails: spotDetailsModalSlice.reducer,
    searchedSpotsListModal: searchedSpotListModalSlice.reducer,
    expandedSpotMediaGallery: expandedSpotMediaGallerySlice.reducer,
    spotFilters: spotFiltersSlice.reducer,
    chats: chatsSlice.reducer,
    map: mapSlice.reducer,
    sidebar: sidebarSlice.reducer,
    searchedSpots: searchedSpotsSlice.reducer,
    social: socialSlice.reducer,
    spotComments: spotCommentSlice.reducer,
    currentViewSpots: currentViewSpotsSlice.reducer,
    currentViewSpotsListModal: currentViewSpotsListModalSlice.reducer,
    currentViewSpotsParams: currentViewSpotParamsSlice.reducer,
    spotWeather: spotWeatherSlice.reducer,
    expandedSpotGalleryMediaList: expandedSpotGalleryMediaListSlice.reducer,
    expandedSpotMediaGalleryModals: [
      expandedSpotMediaGalleryModalsSlice.reducer,
    ],
    expandedSpotMediaGalleryFullscreenSizeModal: [
      expandedSpotMediaGalleryFullscreenSizeSlice.reducer,
    ],
    expandedSpotGalleryCurrentMedia: [
      expandedSpotGalleryCurrentMediaSlice.reducer,
    ],
    spotAddMediaModal: addSpotMediaModalSlice.reducer,
    forum: forumModalSlice.reducer,
    forumReport: forumReportModalSlice.reducer,
  },
},
);

export default store; Show usages ⚡ Mredosz
export type RootState = ReturnType<typeof store.getState>;
export type AppDispatch = typeof store.dispatch;

```

Rysunek 7.8: Konfiguracja sklepu (Redux store)

```

interface AccountSliceProps { Show usages ▾ Mredosz +1
  isLoggedIn: boolean;
  username: string;
}

const initialState: AccountSliceProps = {
  isLoggedIn: localStorage.getItem("is_logged_in") === "true",
  username: localStorage.getItem("username") || "",
};

export const accountSlice : Slice<AccountSliceProps, { setisLoggedIn(st...} = createSlice({ Show usages ▾ Mredosz +1
  name: "account",
  initialState,
  reducers: {
    setIsLoggedIn(state : WritableDraft<AccountSliceProps>) : void {
      localStorage.setItem("is_logged_in", "true");
      state.isLoggedIn = true;
    },
    signOut(state : WritableDraft<AccountSliceProps>) : void {
      localStorage.removeItem("is_logged_in");
      localStorage.removeItem("username");
      state.isLoggedIn = false;
      state.username = "";
    },
    setUsername(state : WritableDraft<AccountSliceProps>, action: PayloadAction<string>) : void {
      localStorage.setItem("username", action.payload);
      state.username = action.payload;
    },
  },
},
);

export const accountAction : CaseReducerActions<{ setisLoggedIn(state: W...} = accountSlice.actions; Show usages ▾ Mredosz

```

Rysunek 7.9: Przykładowy slice odpowiedzialny za sprawdzenie czy użytkownik jest zalogowany

### 7.2.3 Integracja i komunikacja z backendem

W niniejszym podrozdziale opisano sposób integracji aplikacji [frontendowej](#) z [backendem](#) oraz mechanizmy odpowiedzialne za bezpieczną i efektywną komunikację z serwerem.

Jest to kluczowy element aplikacji, ponieważ wymaga bezpiecznego przesyłania danych użytkownika. W celu uproszczenia komunikacji z serwerem zdecydowano się na wykorzystanie biblioteki [axios](#) [16] oraz [biblioteki TanStack Query](#) [17]. We

wszystkich ścieżkach wymagających zalogowania użytkownika do zapytania dołączany jest token **JWT**. Token przekazywany jest w ciasteczku dzięki ustawieniu parametru **withCredentials** na wartość **true**. Przykładem pliku odpowiedzialnego za taką komunikację jest **account.js** (rys. 7.10 i 7.11), który obsługuje operacje związane z logowaniem/rejestracją, zmianą hasła oraz wylogowaniem.

```
export async function loginUser(user) { Show usages △ Adam Langmesser +1
  return await axios.post(`${BASE_URL}/public/account/login`, user, {
    withCredentials: true,
  });
}

export async function registerUser(user) { Show usages △ Mredosz +2
  return await axios.post(`${BASE_URL}/public/account/register`, user, {
    withCredentials: true,
  });
}

export async function sentEmailWithNewPasswordLink(email) { Show usages △ Adam Langmesser +1 *
  return await axios.post(
    `${BASE_URL}/public/account/forgot-password`,
    email,
    {
      headers: {
        "Content-Type": "text/plain",
      },
    },
  );
}
```

Rysunek 7.10: Implementacja modułu account (1)

```

export async function changePassword(userData) { Show usages  ↳ stanoz +1
  return await axios.post(
    `${BASE_URL}/public/account/set-new-password`,
    userData,
  );
}

export async function logout() { Show usages  ↳ stanoz +1
  await axios.post(
    `${BASE_URL}/account/oauth2/logout`,
    {},
    {
      withCredentials: true,
    },
  );
}

export const googleLoginUrl = `${BASE_URL}/oauth2/authorization/google`; Show usages  ↳ stanoz
export const githubLoginUrl = `${BASE_URL}/oauth2/authorization/github`; Show usages  ↳ stanoz

```

Rysunek 7.11: Implementacja modułu account (2)

Funkcje odpowiedzialne za komunikację z backendem umieszczone w katalogu `/http`. Dzięki temu są one skoncentrowane i mogą być w prosty sposób wykorzystywane w różnych częściach aplikacji. Zastosowanie TanStack Query umożliwiło znaczące ograniczenie powtarzalnego kodu oraz uprościło obsługę błędów i stanów zapytania (takich jak ładowanie danych, błąd czy sukces). Biblioteka udostępnia m.in. wartość `isLoading`, dzięki czemu komponent może łatwo wyświetlić ekran ładowania bez konieczności ręcznego zarządzania własnym stanem. Dodatkowo `hook useQuery` pozwala na automatyczne pobieranie danych po wejściu na daną podstronę. Komponent deklaruje jedynie, jakie dane są mu potrzebne, a TanStack Query realizuje ich pobranie, cache'owanie oraz odświeżanie. Do operacji wymagających wywołania akcji po stronie użytkownika (np. wysłania formularza logowania) wykorzystywany jest `hook useMutation` z TanStack Query. Przykład użycia tego rozwiązania w procesie logowania przedstawiono na rys. 7.12.

```
const { mutateAsync, isSuccess, error } = useMutation({
  mutationFn: loginUser,
});

const handleSubmit : (event: FormEvent<HTMLFormElement>) => Promise<void> = async (event) : Promise<void> => {
  event.preventDefault();
  await mutateAsync({
    username: enteredValue.username,
    password: enteredValue.password,
  });
  navigate(-1);
};
```

Rysunek 7.12: Wykorzystanie TanStack Query przy logowaniu użytkownika

#### 7.2.4 Style

W niniejszym podrozdziale przedstawiono zastosowane w projekcie rozwiązania dotyczące stylowania interfejsu użytkownika oraz narzędzia wykorzystywane do tworzenia spójnej i **responsywnej** warstwy wizualnej aplikacji.

Do stylowania interfejsu wykorzystano **framework** Tailwind CSS [18]. Dzięki gotowym klasom udostępnianym przez Tailwind wygląd elementów można definiować bezpośrednio w kodzie komponentu, bez konieczności przechodzenia do osobnych plików ze stylami. Ułatwia to zarówno tworzenie widoków, jak i późniejsze modyfikacje — w przypadku zmiany stylu dokładnie wiadomo, gdzie należy jej dokonać. Korzystanie ze zdefiniowanych klas pozwoliło zachować spójność wizualną w całej aplikacji. W pliku `index.css` zdefiniowano zmienne kolorystyczne (rys. 7.13 i 7.14). Dzięki temu zmiana motywów kolorystycznego w przyszłości sprowadza się do edycji wartości w jednym miejscu.



```
--height-1\10: 10%;  
--breakpoint-3xl: 160rem;  
--color-mainBlue: #4242f0;  
--color-mainBlueDarker: #0d0db5;  
--color-darkText: #e5e5e5;  
--color-darkBg: #0f0f10;  
--color-darkBgSoft: #1b1c1d;  
--color-grayBg: #d9d9d9;  
--color-darkBgMuted: #323539;  
--color-darkBorder: #939394;  
--color-lightText: #222222;  
--color-lightBg: #e4e3e3;  
--color-lightBgDarker: #cccaca;  
--color-lightBgSoft: #ffffff;  
--color-lightBgMuted: #f2f2f2;  
--color-lightBorder: #fbfdff;  
--color-lightGrayishViolet: #f2eef9;  
--color-whiteSmoke: #f6f6f6;  
--color-warmerWhiteSmoke: #ece9e9;  
--color-lightGrayishBlue: #e5e9ee;  
--color-paleBlueGray: #acafbb;  
--color-grayText: #d3d3d3;
```

Rysunek 7.13: Implementacja zmiennych kolorystycznych (1)



```
--color-violetDark: #363041;
--color-violetLight: #6d6183;
--color-violetLightDarker: #4f4660;
--color-violetLightDark: #554a69;
--color-violetLighter: #9b8cbd;
--color-violetDarker: #2c2734;
--color-violetHeavyDark: #1e1b23;
--color-violetBtnBorderDark: #625b6e;
--color-violetBright: #835ace;
--color-darbVioletBtnOutline: #816ba6;
--color-mediumDarkBlue: #424b77;
--color-first: #2c3e50;
--color-second: #34495e;
--color-third: #1abc9c;
--color-fourth: #16a085;
--color-fifth: #ecf0f1;
--color-sixth: #e94560;
--color-magenta: #a01bc1;
--color-darkYellow: #c5a03c;
--color-ratingStarColor: #fadb14;
--color-locationMarkerDarkerBlue: #a3dcff;
--color-locationMarkerLightBlue: #52bafb;
--color-userLocationDot: #4285f4;
--color-spotLocationMarker: #a8071a;
```

Rysunek 7.14: Implementacja zmiennych kolorystycznych (2)

W niektórych miejscach konieczne było zapisanie stylów w czystym **CSS**, ponieważ część użytych **bibliotek** tego wymagała. W innych przypadkach wystarczyło skorzystać z klas zdefiniowanych w **index.css** oraz klas Tailwinda. Część aplikacji jest **responsywna**. Tailwind udostępnia predefiniowane prefiksy **responsywne** (np. **md:**, **lg:**) (rys. 7.15), utworzono również własny (**3xl:**) na ekrany o rozdzielczości 2560px. Pozwalają one przypisywać style zależnie od szerokości ekranu bez pisania własnych reguł **media queries**. Dzięki temu implementacja widoków mobilnych i desktopowych była znaczco szybsza.

```
<div className="mt-17 flex flex-col items-center gap-7 lg:mt-0 lg:-ml-40 lg:flex-row xl:-ml-42 xl:gap-10 2xl:-ml-80">
  <div className="relative">
    <img alt="profileImage"
      src={userData?.profilePhoto}
      className="dark:drop-shadow-darkBgMuted aspect-square h-64 rounded-full
      shadow-md sm:h-80 lg:h-85 xl:h-96 dark:drop-shadow-md"
    />
```

Rysunek 7.15: Przykładowe użycie klas Tailwind (w tym prefiksów responsywności)

Tailwind został też wykorzystany do obsługi trybu jasnego i ciemnego. Wystarczy dodać klasę z prefiksem **dark:** (np. **dark:bg-black**), aby zmienić kolorystykę elementu, gdy aplikacja jest w trybie ciemnym (rys. 7.16).

```
<input
  id={id}
  value={value}
  type={type}
  onChange={onChange}
  onFocus={setFocusedToTrue}
  onBlur={handleOnBlur}
  className="dark:bg-darkBgMuted bg-lightBgMuted dark:text-darkText text-lightText w-full rounded-md
  p-2 shadow-md focus:outline-none dark:shadow-black/50"
/>
```

Rysunek 7.16: Przykładowe użycie klas Tailwind (w tym wariantu **dark:**)

Aby uzyskać płynniejsze i przyjemniejsze animacje, wykorzystano **bibliotekę Motion [19]**. Pozwala ona w prosty sposób tworzyć animacje elementów interfejsu, bez potrzeby ręcznego pisania złożonych reguł **CSS**. W aplikacji wykorzystano ją

m.in. w polach formularza logowania i rejestracji (rys. 7.17). Na początku etykieta pola (np. „username”) jest wyświetlana wewnątrz pola tekstowego, natomiast po kliknięciu w pole jest płynnie przesuwana nad to pole, co poprawia czytelność i ergonomię formularza.

```
<motion.label
  htmlFor={id}
  initial={false}
  animate={{
    top: shouldFloat ? "-0.7rem" : "0.5rem",
    left: "0.75rem",
    fontSize: shouldFloat ? "0.75rem" : "1rem",
    opacity: shouldFloat ? 1 : 0.6,
  }}
  transition={{ type: "spring", stiffness: 300, damping: 25 }}
  className="■ dark:text-darkText ■ text-lightText pointer-events-none absolute z-10 px-1 capitalize"
>
  {label}
</motion.label>
<input
  id={id}
  value={value}
  type={type}
  onChange={onChange}
  onFocus={setFocusedToTrue}
  onBlur={handleOnBlur}
  className="■ dark:bg-darkBgMuted ■ bg-lightBgMuted ■ dark:text-darkText ■ text-lightText w-full rounded-md
  p-2 shadow-md focus:outline-none dark:shadow-black/50"
/>
```

Rysunek 7.17: Implementacja animacji z wykorzystaniem Motion

### 7.2.5 Wyszukiwarka spotów

W niniejszym rozdziale przedstawiono sposób implementacji wyszukiwarki spotów.

Jednym z głównych modułów aplikacji jest wyszukiwarka spotów, umożliwiająca szybkie odnalezienie interesujących lokalizacji. Funkcjonuje ona w dwóch wariantach: prostym i zaawansowanym (rys. 7.18 oraz 7.19).

```
<div className="■dark:bg-darkBg ■dark:text-darkText ■bg-lightBg ■text-lightText  
flex min-h-screen w-full flex-col items-center space-y-4 overflow-hidden p-8 pt-18">  
  <Switch />  
  <SearchBar  
    onSetSpots={handleSetSearchedSpots}  
    loadMoreRef={loadMoreRef}  
    onSetFetchingNextPage={setIsFetchingNextPage}  
  />  
  <div className="flex w-full flex-col items-center space-y-4">  
    <h1 className="text-center text-3xl">The Most Popular Spots</h1>  
    <div className="flex w-full flex-col items-center space-y-5">  
      <Carousel spots={data!} spotsPerPage={spotsPerPage} />  
      <SearchSpotList  
        spots={searchedSpots}  
        isFetchingNextPage={isFetchingNextPage}  
        loadMoreRef={loadMoreRef}  
      />  
    </div>  
  </div>  
</div>
```

Rysunek 7.18: Implementacja prostej wersji wyszukiwarki

```
<div className="■dark:bg-darkBg ■dark:text-darkText ■bg-lightBg ■text-lightText  
flex min-h-screen w-full flex-col items-center space-y-4 overflow-hidden p-8 pt-18">  
  <Switch />  
  <AdvanceSearchBar  
    onSetSpots={handleSetSearchedSpots}  
    loadMoreRef={loadMoreRef}  
    onSetFetchingNextPage={setIsFetchingNextPage}  
  />  
  <div className="flex w-full flex-col items-center space-y-10">  
    <SearchSpotList  
      spots={searchedSpots}  
      loadMoreRef={loadMoreRef}  
      isFetchingNextPage={isFetchingNextPage}  
    />  
  </div>  
</div>
```

Rysunek 7.19: Implementacja zaawansowanej wersji wyszukiwarki

Przełączanie pomiędzy tymi widokami odbywa się za pomocą przycisku umieszczonego w górnej części strony (rys. 7.20).

```
<div className="■dark:shadow-darkBgSoft flex rounded-full shadow-lg shadow-black/20">
  <NavLink
    to="/"
    className={({ isActive } : NavLinkRenderProps) : string =>
      `■hover:dark:bg-violetDark ■hover:bg-violetLight rounded-l-full px-2.5 py-1.5
       transition-all duration-300 ${isActive ? "■dark:bg-violetDark ■bg-violetLight" : ""}`}
    }
  >
    Simple filters
  </NavLink>
  <NavLink
    to="/advanced"
    className={({ isActive } : NavLinkRenderProps) : string =>
      `■hover:dark:bg-violetDark ■hover:bg-violetLight rounded-r-full px-2.5 py-1.5
       transition-all duration-300 ${isActive ? "■dark:bg-violetDark ■bg-violetLight" : ""}`}
    }
  >
    Advanced filters
  </NavLink>
</div>
```

Rysunek 7.20: Implementacja komponentu do przełączania trybów

W trybie prostym prezentowana jest karuzela (rys. 7.21) z dwunastoma najpopularniejszymi [spotami](#) w całej aplikacji. W tym widoku możliwe jest wyszukiwanie [spotów](#) po lokalizacji (kraj, region, miasto).

```

<div className="relative flex w-full items-center justify-center">
  <button
    onClick={() : void => paginate(-1)}
    className="hover:text-darkBorder z-10 cursor-pointer transition-all duration-300"
  >
    <RiArrowLeftWideFill className="text-5xl sm:text-6xl" />
  </button>

  <div className="relative h-[440px] w-full max-w-[1200px] overflow-hidden">
    <AnimatePresence custom={direction} initial={false} mode="sync">
      <motion.div
        key={page}
        custom={direction}
        variants={sliderVariants}
        initial="incoming"
        animate="active"
        exit="exit"
        transition={[ 3 elements... ]}
        className="grid w-full grid-cols-1 grid-rows-1 justify-items-center gap-4
          lg:grid-cols-2 lg:grid-rows-2 2xl:grid-cols-3 2xl:grid-rows-2"
      >
        {currentSpots.map((spot : TopRatedSpot) : Element => (
          <MostPopularSpot
            spot={spot}
            key={`${spot.id}-${page}`}
          />
        ))}
      </motion.div>
    </AnimatePresence>
  </div>

  <button
    onClick={() : void => paginate(1)}
    className="hover:text-darkBorder z-10 cursor-pointer transition-all duration-300"
  >
    <RiArrowRightWideFill className="text-5xl sm:text-6xl" />
  </button>
</div>

```

Rysunek 7.21: Implementacja karuzeli z najpopularniejszymi [spotami](#)

Widok zaawansowany udostępnia rozszerzoną wyszukiwarkę, która umożliwia filtrowanie wyników po mieście, tagach oraz ocenie, a także ich sortowanie według popularności i średniej oceny (rys. 7.19).

Wyszukiwarka spotów została zbudowana z dwóch głównych komponentów: `HomePage` oraz `AdvanceHomePage`. W skład prostej wersji wchodzą następujące komponenty:

- `Switch` – służy do przełączania widoku między trybem podstawowym a zaawansowanym,
- `SearchBar` – podstawowa wyszukiwarka `spotów`,
- `Carousel` – wyświetla najpopularniejsze `spotty`,
- `SearchSpotList` – wyświetla wyszukane `spotty`.

W skład zaawansowanej wersji wchodzą następujące komponenty:

- `Switch` – służy do przełączania widoku między trybem podstawowym a zaawansowanym,
- `AdvanceSearchBar` – zaawansowana wyszukiwarka `spotów`,
- `SearchSpotList` – wyświetla wyszukane `spotty`.

Komponent `Switch` (rys. 7.20) zawiera dwa elementy `NavLink` z biblioteki React Router, co pozwala na przełączanie widoków bez konieczności przeładowywania całej strony.

W komponencie `SearchBar` (rys. 7.22) po wpisaniu co najmniej dwóch znaków wyświetlana jest lista podpowiedzi dla kraju, regionu oraz miasta, w zależności od aktualnie uzupełnianego pola. Po pojawienniu się listy możliwe jest wybranie odpowiedniej lokalizacji, co ułatwia określenie lokalizacji dostępnych `spotów`.

```

<div className="■dark:bg-darkBgSoft □bg-lightBgSoft flex w-full flex-col items-center justify-between space-y-3 rounded-md px-3 py-2 shadow-md md:flex-row md:space-y-0 lg:w-3/4 lg:space-x-3 xl:w-1/2 ■dark:shadow-black">
  <div className="flex w-full flex-col space-y-2">
    <h1>Location</h1>
    <div className="flex w-full flex-col space-y-3 md:flex-row md:space-y-0 md:space-x-2">
      {inputList.map(({ id, label }) :{readonly label: "Your Country"; readonly id: string} : Element => (
        <div key={id} className="relative w-full">
          <SearchInput
            label={label}
            id={id}
            value={searchLocation[id] ?? ""}
            onChange={(e : ChangeEvent<HTMLInputElement>} : void =>
              handleSetLocation(id, e.target.value)
            }
            onFocus={() : void => setActiveInput(id)}
          />
          {activeInput === id && suggestions.length > 0 && (
            <SearchSuggestions
              suggestions={suggestions}
              onClick={handleSuggestionClick}
              id={id}
              onClose={() : void => setActiveInput(null)}
            />
          )}
        </div>
      )));
    </div>
    <button
      className="■dark:bg-darkBgMuted ■dark:hover:bg-darkBgMuted/80 □bg-lightBgMuted
      □hover:bg-lightBgMuted/80 flex w-full cursor-pointer justify-center rounded-md p-2 md:w-fit"
      onClick={handleSearchSpots}
    >
      <FaSearch />
    </button>
  </div>
</div>

```

Rysunek 7.22: Implementacja prostej wyszukiwarki

Komponent `SearchSpotList` (rys. 7.23) odpowiada za prezentację wyników wyszukiwania. Został w nim zaimplementowany mechanizm przewijania nieskończonego ([infinite scroll](#)), który automatycznie pobiera kolejne strony wyników w momencie, gdy użytkownik zbliża się do końca listy. Wykorzystuje on listę komponentów `SpotTile`, a także komponent `LoadingSpinner` oraz komunikat informujący o braku wyników, jeżeli nie zostanie odnaleziony żaden `spot`.

```

<>
  <ul className="grid w-full grid-cols-1 place-items-center gap-8 xl:grid-cols-2 2xl:grid-cols-3">
    {spots.map((spot : HomePageSpotDto) : Element => (
      <SpotTile key={spot.id} spot={spot} />
    )))
  </ul>
  <div ref={loadMoreRef} className="h-10" />
  {isFetchingNextPage && <LoadingSpinner />}
  {spots.length === 0 && (
    <p className="text-center text-2xl">
      | Search for spots to see results.
    </p>
  )}
</>

```

Rysunek 7.23: Implementacja listy do wyświetlania **spotów**

Komponent **SpotTile** zawiera następujące informacje:

- zdjęcie **spota**,
- miasto, w którym się znajduje,
- nazwę **spota**,
- ocenę oraz liczbę ocen,
- tagi,
- podstawowe informacje pogodowe (temperatura i typ pogody),
- dwa przyciski: jeden prowadzący do widoku szczegółów **spota** oraz drugi informujący, jak daleko znajduje się dany **spot**; po kliknięciu przycisku prezentowana jest lokalizacja **spota** na mapie.

Komponent **AdvanceSearchBar** jest zbliżony wyglądem i strukturą do wersji podstawowej, jednak w polu lokalizacji można podać wyłącznie miasto. Dodatkowo dostępna jest możliwość dodawania tagów z przygotowanej listy. Wyszukiwarka umożliwia także filtrowanie po ocenie oraz sortowanie wyników według oceny i popularności z wykorzystaniem komponentów typu **Dropdown**.

Oba widoki (`HomePage` i `AdvanceHomePage`) współdzielą część komponentów, między innymi `Switch` oraz `SearchSpotList`. Dzięki temu kod odpowiedzialny za wyświetlanie listy wyników jest zdefiniowany w jednym miejscu, a zmiany w sposobie prezentacji `spotów` wymagają modyfikacji tylko w komponentach wspólnie dzielonych.

#### **7.2.6 Mapa**

#### **7.2.7 Chat**

#### **7.2.8 Forum**

#### **7.2.9 Konto użytkownika**

#### **7.2.10 Panel logowania**

### **7.3 Implementacja CI/CD**

# Rozdział 8

## Testy

- 8.1 Testy jednostkowe
- 8.2 Testy integracyjne
- 8.3 Testy E2E
- 8.4 Wyniki testów i wnioski

## Rozdział 9

### Prezentacja systemu

9.1 Strona główna

9.2 Strona mapy

9.3 Strona chatu

9.4 Strona forum

9.5 Panel logowania

9.6 Panel konta użytkownika

# Rozdział 10

## Nakład pracy

### 10.1 Ogólny nakład pracy

### 10.2 Indywidualne nakłady pracy

#### 10.2.1 Adam Langmesser

#### 10.2.2 Mateusz Redosz

Na projekt poświęciłem łącznie 324 godziny, z czego 237 przeznaczyłem na prace deweloperskie, 111 na pisanie dokumentacji, 19 godzin na [Review kodu](#), 19 na spotkania dotyczące omówienia dalszych prac projektowych oraz przy pomocy innym członkom zespołu oraz 49 godzin poświęciłem nad stworzeniem widoków na figmie. Prace nad częścią deweloperską rozpoczęłem 04.08.2024 a zakończyłem 08.09.2025. W projekcie pracowałem nad Rejestracją użytkownika, tokenem [JWT](#), częściową implementacją [CI/CD](#), stroną główną, zaimplementowaniem [Sidebara](#) oraz podstroną dla użytkownika. Moje wylistowane zadania z Jira:

1. Dokumentacja
  - TODO
2. Design
  - Ustalić paletę kolorystyczną

- Propozycja wyglądu

### 3. Backend i Frontend

- Formularz rejestracji
- Routing
- Formatowanie w React (prettier)
- Obsługa JWT na frontend
- oAuth Frontend
- Update JWT
- Refactor JWT
- Stworzenie komponentu Notification i poprawa błędów
- Implementacja pierwszych testów
- Zaimplementowanie kolejki w komponencie notification
- Dodanie reduxa do rejestracji
- Zmiana sposobu pobierania danych o spotach
- Obsługa customowych błędów z jakarta.validation
- Obsługa auto wylogowania przy starcie
- Domyślna wiadomość w notification
- Poprawa headera
- Ciemny motyw
- Refactor pogody
- Propozycja wyglądu
- Przeniesienie zdjęć z google drive
- Dodać Type script do Reacta
- Aktualizacja tailwinda i dodanie kolorów
- Podstawowy Sidebar

- Strona główna z prostymi filtrami
- Strona główna z zaawansowanymi filtrami
- [Sidebar](#)
- Strona profilu
- Ustawienia
- Listy spotów
- Lista zdjęć
- Lista filmów
- Lista znajomych
- Dodanie spotów
- Lista komentarzy
- Strona główna profilu
- Listy
- Poprawa [Sidebara](#)
- Zmiana kropki na przyciemnienie tła na [Sidebar](#)
- Poprawa strony do logowania i rejestracji
- Usunięcie username z account Redux
- Dodanie zamknięcia [Sidebara](#) na małych ekranach po kliknięciu nav linka
- Poprawić tooltipa na sidebar
- Zmiana sposobu pobierania username na backendzie z tokena jwt
- Paginacja z infinity scrolllem
- Lista zdjęć innego usera
- Walidacja i responsywność w dodaniu spotów
- Dodanie sortowania i filtrów na zaawansowanej stronie
- Zmiana na infinity scrola

- Zmiana zdjęcia profilowego użytkownika
- Czyszczenie formularza w dodawaniu spota
- Dodanie wyszukiwarki znajomych w Social
- Zatwierdzenie przez drugiego użytkownika dodania do znajomych
- Sprawdzenie czy wszystko działa i poprawki Mateusz

#### 4. CI/CD

- Dodanie testów z frontendu do github actions
- Poprawa github actions
- Poprawa pipeline od Javy i Reacta

#### 5. Praca dyplomowa

- Uzupełnienie informacji o zespole i podział na rozdziały

### **10.2.3 Stanisław Oziemczuk**

### **10.2.4 Kacper Badek**

# Rozdział 11

## Podsumowanie

11.1 Osiągnięte rezultaty

11.2 Napotkane wyzwania

11.3 Plany na przyszłość

# Rozdział 12

## Słownik pojęć i skrótów

### **API**

(ang. *application programming interface*); zbiór reguł i operacji do komunikacji z oprogramowaniem.. [17](#), [18](#), [88](#)

### **Backend**

Część aplikacji odpowiedzialna za logikę biznesową, przetwarzanie danych i komunikację z bazą danych. Działa po stronie serwera i obsługuje żądania wysyłane przez frontend. [2](#), [15](#), [115](#), [116](#), [118](#), [121](#), [170](#), [181](#), [199](#)

### **Backlog**

Lista zadań, które należy wykonać w ramach projektu, używane w metodykach zwinnych.. [16](#)

### **Baza danych**

Zbiór uporządkowanych danych przechowywanych w sposób umożliwiający ich łatwe wyszukiwanie, modyfikowanie i analizowanie. W aplikacjach najczęściej wykorzystywane są relacyjne lub nierelacyjne bazy danych. [82](#), [84](#), [86–88](#), [115](#), [116](#), [118](#), [170](#)

### **Biblioteka**

Zewnętrzny lub wewnętrzny zestaw gotowych funkcji, klas, komponentów lub modułów, który można wielokrotnie wykorzystywać w projekcie zamiast pisać wszystko od zera. [116](#), [170](#), [176](#), [178](#), [181](#), [187](#), [192](#)

## **BPMN**

(ang. *Business Process Model and Notation*); standardowa notacja graficzna, która umożliwia szczegółowe przedstawienie i dokumentowanie procesów biznesowych..

[18](#)

## **Cache**

Mechanizm przechowywania danych w celu przyspieszenia ich ponownego odczytu.

[115](#), [116](#)

## **CI/CD**

Skrót od *Continuous Integration/Continuous Deployment*. Praktyka programistyczna polegająca na automatyzacji procesu budowania, testowania i wdrażania oprogramowania. [16](#), [198](#), [201](#)

## **CSS**

Kaskadowe arkusze stylów (Cascading Style Sheets) — język opisu prezentacji dokumentów (np. HTML). Definiuje wygląd interfejsu: układ, kolory, typografię, odstępy, animacje i zachowania responsywne, oddzielając warstwę treści od warstwy prezentacji.. [187](#)

## **Design**

Etap lub proces projektowania wyglądu i funkcjonalności aplikacji, obejmujący zarówno aspekty wizualne, jak i użytkowe (UX/UI). [198](#)

## **Disciplined Agile Delivery - Lean Life Cycle**

Disciplined Agile Delivery w wariantie Lean Life Cycle to sposób prowadzenia projektu, który łączy elastyczność Agile z przewidywalnością Waterfalla, ale bez stałych sprintów — praca toczy się w ciągłym przepływie. Na starcie zakłada mocniejszą fazę przygotowawczą: doprecyzowanie zakresu, szkic architektury, identyfikację ryzyk i kryteria jakości. W realizacji następuje ciągłe doprecyzowywanie wymagań i backlogu, oparte na regularnym feedbacku udziałowców. Całość opiera

się na praktykach Lean oraz lekkim governance: code review i regularnych przeglądach postępów. . [11](#)

## **Droniarz**

Potoczne określenie osoby, która jest jednocześnie pilotem oraz operatorem drona. Zwykle entuzjasta dronów.. [9](#), [10](#), [211](#)

## **Droniarz foto/video**

Pilot wykorzystujący drony fotograficzne/filmowe do rejestracji materiałów wizualnych (zdjęcia, wideo), zwykle z naciskiem na stabilizację i jakość obrazu.. [18](#)

## **emoji**

Małe graficzne ikonki używane do wyrażania emocji lub pojęć w komunikacji cyfrowej (np. uśmiechnięta buźka, kciuk w górę, symbol serca).. [82](#), [83](#)

## **Folder by type**

Sposób organizowania struktury katalogów w projekcie, w którym pliki są grupowane według rodzaju (typu) zasobu, a nie według funkcjonalności. Na przykład wszystkie komponenty trafiają do jednego folderu, wszystkie style do innego itd. [121](#), [173](#)

## **Framework**

Zestaw narzędzi, bibliotek i struktur wspomagających tworzenie aplikacji. Ułatwia programowanie poprzez dostarczenie gotowych komponentów oraz określenie zasad organizacji kodu. [2](#), [116](#), [121](#), [184](#)

## **Frontend**

Warstwa aplikacji odpowiedzialna za interfejs użytkownika oraz interakcję z użytkownikiem. Zazwyczaj tworzona przy użyciu technologii takich jak HTML, CSS i JavaScript. [2](#), [15](#), [115](#), [116](#), [118](#), [173](#), [181](#), [199](#)

## **GIF**

Format graficzny *Graphics Interchange Format* obsługujący krótkie, zapętlone animacje. W aplikacjach czatowych wykorzystywany do wysyłania „reakcji” w postaci ruchomych obrazków. [77](#), [78](#)

## **Hook (React)**

Prosta funkcja w React, która „dodaje” możliwości do elementu interfejsu — np. pozwala mu coś zapamiętać (stan) albo zrobić coś po zmianie/załadowaniu. Wszystkie hooki zaczynają się od `use...` (np. `useState`, `useEffect`).. [178](#), [179](#), [183](#)

## **IDE**

(ang. *integrated development environment*); to zintegrowane środowisko programistyczne, służące do tworzenia, modyfikowania, testowania i konserwacji oprogramowania. [15](#)

## **Infinite scroll**

Wzorzec interfejsu użytkownika, w którym kolejne porcje treści są automatycznie doładowywane podczas przewijania strony w dół, zamiast być podzielone na odrębne, ręcznie przełączane strony. [88](#), [193](#)

## **JWT**

Skrót od *JSON Web Token*. Standard służący do bezpiecznego przekazywania informacji między stronami w formacie JSON, często używany w procesach autoryzacji użytkowników. [182](#), [198](#)

## **Media queries**

Konstrukcja CSS pozwalająca stosować reguły stylów w zależności od cech urządzenia/okna (np. szerokości ekranu, orientacji, preferencji użytkownika). Podstawa responsywnego projektowania (*responsive design*).. [187](#), [208](#)

## **paginacja**

Mechanizm dzielenia dużych zbiorów danych (np. list postów, wyników wyszukiwania, komentarzy) na mniejsze strony, które są pobierane i wyświetlane stopniowo, zamiast ładowania wszystkich elementów jednocześnie.. [88](#)

## **PANSA**

Polish Air Navigation Services Agency, pol. Polska Agencja Żeglugi Powietrznej. Instytucja ta zapewnia m.in. mapę z zaznaczonymi strefami lotów. Każda strefa ma swoje właściwości prawne. . [23](#), [42](#), [43](#), [212](#)

## **Parametry zapytania (query params)**

Pary **klucz=wartość** przekazywane w części adresu URL po znaku zapytania ?, służące m.in. do filtrowania, sortowania, paginacji wyników lub przekazywania dodatkowych opcji żądania do serwera, np. ?param1=val1&param2=val2. [124](#), [130–140](#), [142](#), [144](#), [146–149](#), [151–154](#), [156–164](#), [166–170](#)

## **Props**

Właściwości przekazywane do komponentu React przez komponent nadzędny; służą do konfiguracji i przekazywania danych. Powinny być traktowane jako tylko do odczytu (read-only) wewnątrz komponentu potomnego.. [178](#)

## **Protected route**

Trasa w aplikacji, do której dostęp jest ograniczony, zwykle tylko dla zalogowanych użytkowników lub użytkowników z odpowiednimi uprawnieniami. Jeżeli użytkownik nie spełnia warunków, jest przekierowywany (np. na stronę główną). [177](#)

## **React**

Biblioteka JavaScript do budowy interfejsów użytkownika w oparciu o komponenty deklaratywne i wirtualny DOM. Zapewnia jednokierunkowy przepływ danych oraz zarządzanie stanem komponentów.. [115](#), [116](#), [178](#)

## Redis

Szybka baza danych typu klucz-wartość przechowywana głównie w pamięci operacyjnej. Często wykorzystywana jako pamięć podręczna (cache), magazyn sesji lub prosty mechanizm komunikatów między usługami. [115](#), [116](#), [118](#)

## Redux

Biblioteka do przewidywalnego zarządzania stanem aplikacji. Opiera się na jednym *store*, akcjach i czystych *reducerach*, promuje niemutowalność i jednokierunkowy przepływ danych. Często używana z Reactem, ale niezależna od niego.. [178](#), [179](#)

## Responsywność

Określenie związane z projektowaniem responsywnym (Responsive Web Design, RWD), czyli dostosowywaniem interfejsu do różnych rozmiarów i parametrów ekranów. Obejmuje m.in. elastyczne siatki, grafiki i [Media queries](#), tak aby układ i czytelność były zachowane na telefonach, tabletach i desktopach.. [184](#), [187](#)

## REST API

Architektura budowania usług sieciowych komunikujących się poprzez metody protokołu HTTP (GET, PUT, POST, DELETE, PATCH). Wymiana danych występuje często w formacie JSON lub XML.

REST API musi spełniać następujące reguły:

1. **Rozdzielenie klient-serwer** — klient i serwer są od siebie niezależne, komunikują się poprzez interfejs.
2. **Bezstanowość** — każde żądanie przez klienta zawiera wszystkie informacje niezbędne do jego obsłużenia. Po otrzymaniu żądania serwer nie przechowuje o nim żadnych informacji.
3. **Buforowalność (cache)** — odpowiedzi z API powinny informować, czy dane można cache'ować. Jeśli tak, to przy kolejnym żądaniu mogą być zwrócone z cache'a.
4. **Jednolity interfejs:**
  - **Identyfikacja zasobów** — każdy zasób musi być jednoznacznie zidentyfikowany w interakcji klient-serwer.

- **Manipulacja zasobów poprzez reprezentację** — po otrzymaniu reprezentacji klient może zmienić stan zasobu przesyłając zmodyfikowaną reprezentację.
  - **Samoopisujące się wiadomości** — każde żądanie i odpowiedź powinny zawierać informacje do jego poprawnego przetworzenia.
  - **Hypermedia jako silnik stanu aplikacji (HATEOAS)** — po otrzymaniu odpowiedzi klient powinien móc dynamicznie poznać inne interakcje przez linki.
5. **Warstwowość** — klient nie wie czy komunikuje się bezpośrednio z serwrem, czy poprzez pośrednika (np. proxy) oraz nie wie z czym komunikuje się obsługująca go warstwa.
  6. **Kod na żądanie (opcjonalnie)** — serwer może przesłać fragment kodu, który zostanie wykonany przez klienta.

[16](#), [116](#), [121](#), [124](#)

## Review kodu

Proces polegający na wzajemnym przeglądzie kodu źródłowego przez programistów w celu wykrycia błędów, poprawy jakości oraz zwiększenia spójności projektu. [16](#), [17](#), [198](#)

## Sidebar

Boczny panel w interfejsie użytkownika, zawierający menu nawigacyjne lub dodatkowe opcje funkcjonalne aplikacji. [104](#), [198–200](#)

## Spot

Potencjalne miejsce do latania dronem, zaznaczone na mapie.. [16](#), [76](#), [190–195](#)

## Stan

Aktualny zestaw danych przechowywanych przez aplikację lub komponent, na podstawie którego renderowany jest interfejs użytkownika. Stan może być lokalny (utrzymywany w pojedynczym komponencie) albo globalny (wspólny dla wielu komponentów).. [178](#)

## **Tablica Kanban**

Narzędzie do zarządzania przepływem pracy, które pomaga zespołom śledzić zadania oraz ich postępy. Składa się z kolumn reprezentujących stan etapu prac, na przykład „Do zrobienia” lub „W trakcie”.. [16](#)

## **TypeScript**

Rozszerzenie do języka JavaScript dodający statyczne typowanie, interfejsy i narzędzia do większych projektów. Kompiluje się do czystego JavaScript, ułatwiając wykrywanie błędów w czasie kompilacji i refaktoryzacji.. [115](#), [116](#), [178](#)

## **UI**

Interfejs użytkownika (ang. *User Interface*); warstwa prezentacji odpowiedzialna za sposób wyświetlania danych oraz interakcji użytkownika z aplikacją.. [17](#), [81](#), [84](#), [85](#), [178](#)

## **UML**

(ang. *Unified Modeling Language*); graficzny język wizualizacji, specyfikowania oraz dokumentowania składników systemów informatycznych. . [18](#)

# Spis tabel

Tabela 2.1: Karta udziałowca: Zespół projektowy . . . . .	8
Tabela 2.2: Karta udziałowca: Promotor . . . . .	9
Tabela 2.3: Karta udziałowca: Droniarze . . . . .	10
Tabela 3.1: Usługa zewnętrzna: GitHub Actions (CI) . . . . .	19
Tabela 3.2: Usługa zewnętrzna: Azure Blob Storage . . . . .	19
Tabela 3.3: Usługa zewnętrzna: Mailtrap . . . . .	19
Tabela 3.4: Usługa zewnętrzna: LocationIQ . . . . .	20
Tabela 3.5: Usługa zewnętrzna: Google Maps (Maps URLs) . . . . .	20
Tabela 3.6: Usługa zewnętrzna: OpenFreeMap . . . . .	20
Tabela 3.7: Usługa zewnętrzna: Open-Meteo . . . . .	21
Tabela 3.8: Usługa zewnętrzna: Tenor GIF API . . . . .	21
Tabela 3.9: Usługa zewnętrzna: Where the ISS at? . . . . .	21
Tabela 4.1: Scenariusz przypadku użycia: Rejestracja użytkownika . . . . .	34
Tabela 4.2: Scenariusz przypadku użycia: Logowanie użytkownika . . . . .	35
Tabela 4.3: Scenariusz przypadku użycia: Wykupienie subskrypcji premium . . . . .	36
Tabela 4.4: Scenariusz przypadku użycia: Resetowanie hasła . . . . .	37
Tabela 4.5: Scenariusz przypadku użycia: Zmiana hasła w ustawieniach konta . . . . .	38
Tabela 4.6: Scenariusz przypadku użycia: Wylogowanie użytkownika . . . . .	39
Tabela 4.7: Scenariusz przypadku użycia: Przeglądanie powiadomień . . . . .	39
Tabela 4.8: Scenariusz przypadku użycia: Przeszukiwanie historii czatu . . . . .	40
Tabela 4.9: Scenariusz przypadku użycia: Przeglądanie wysłanych plików na czacie . . . . .	41

Tabela 4.10: Scenariusz przypadku użycia: Zmiana typu mapy . . . . .	42
Tabela 4.11: Scenariusz przypadku użycia: Przeglądanie stref PANSA . .	43
Tabela 4.12: Scenariusz przypadku użycia: Wyszukiwanie spota w globalnej wyszukiwarce . . . . .	44
Tabela 4.13: Scenariusz przypadku użycia: Przejście do spota na mapie z wyszukiwarki . . . . .	45
Tabela 4.14: Scenariusz przypadku użycia: Przeglądanie mapy spotów . .	45
Tabela 4.15: Scenariusz przypadku użycia: Otwarcie szczegółów spota . .	46
Tabela 4.16: Scenariusz przypadku użycia: Przeglądanie komentarzy do spota . . . . .	47
Tabela 4.17: Scenariusz przypadku użycia: Przeglądanie pogody na spocie	48
Tabela 4.18: Scenariusz przypadku użycia: Wyszukiwanie spota na mapie	49
Tabela 4.19: Scenariusz przypadku użycia: Utworzenie prywatnego czatu	49
Tabela 4.20: Scenariusz przypadku użycia: Otworzenie czatu . . . . .	50
Tabela 4.21: Scenariusz przypadku użycia: Utworzenie czatu grupowego .	51
Tabela 4.22: Scenariusz przypadku użycia: Przeglądanie listy czatów . .	52
Tabela 4.23: Scenariusz przypadku użycia: Wysyłanie wiadomości na czacie	53
Tabela 4.24: Scenariusz przypadku użycia: Wysyłanie GIF-a na czacie .	54
Tabela 4.25: Scenariusz przypadku użycia: Wysyłanie pliku na czacie .	55
Tabela 4.26: Scenariusz przypadku użycia: Edycja ustawień czatu . . . .	56
Tabela 4.27: Scenariusz przypadku użycia: Dodanie członka do czatu grupowego . . . . .	56
Tabela 4.28: Scenariusz przypadku użycia: Przeglądanie postów na forum	57
Tabela 4.29: Scenariusz przypadku użycia: Wyszukiwanie postów na forum	58
Tabela 4.30: Scenariusz przypadku użycia: Dodanie posta na forum . . .	60
Tabela 4.31: Scenariusz przypadku użycia: Dodanie komentarza na forum	60
Tabela 4.32: Scenariusz przypadku użycia: Przeglądanie historii interakcji z postami . . . . .	61
Tabela 4.33: Scenariusz przypadku użycia: Zarządzanie komentarzami na forum . . . . .	62

Tabela 4.34: Scenariusz przypadku użycia: Zgłoszenie komentarza naruszającego regulamin . . . . .	63
Tabela 4.35: Scenariusz przypadku użycia: Zgłoszenie posta na forum . . . . .	64
Tabela 4.36: Scenariusz przypadku użycia: Przeglądanie komentarzy pod postem . . . . .	65
Tabela 4.37: Scenariusz przypadku użycia: Dodanie spota w panelu użytkownika . . . . .	66
Tabela 4.38: Scenariusz przypadku użycia: Przeglądanie profilu użytkownika . . . . .	67
Tabela 4.39: Scenariusz przypadku użycia: Przeglądanie profilu innego użytkownika . . . . .	68
Tabela 4.40: Scenariusz przypadku użycia: Dodanie użytkownika do znamionych . . . . .	69
Tabela 4.41: Scenariusz przypadku użycia: Przeglądanie społeczności . . . . .	69
Tabela 4.42: Scenariusz przypadku użycia: Przeglądanie dodanych spotów . . . . .	70
Tabela 4.43: Scenariusz przypadku użycia: Edycja danych użytkownika . . . . .	71
Tabela 4.44: Scenariusz przypadku użycia: Przeglądanie dodanych zdjęć do spotów . . . . .	72
Tabela 4.45: Scenariusz przypadku użycia: Przeglądanie dodanych filmów do spotów . . . . .	73
Tabela 4.46: Scenariusz przypadku użycia: Przeglądanie dodanych komentarzy do spotów . . . . .	74
Tabela 4.1: Karta wymagania ogólnego dla czatu: Wysyłanie wiadomości na czacie . . . . .	76
Tabela 4.2: Karta wymagania ogólnego dla czatu: Edycja czatu . . . . .	76
Tabela 4.3: Karta wymagania ogólnego dla czatu: Przeglądanie historii czatu . . . . .	77
Tabela 4.4: Karta wymagania ogólnego dla czatu: Tworzenie czatu . . . . .	77
Tabela 4.1: Wymaganie funkcjonalne dla czatu: Wysyłanie GIF-ów . . . . .	78
Tabela 4.2: Wymaganie funkcjonalne dla czatu: Wysyłanie plików . . . . .	79

Tabela 4.3: Wymaganie funkcjonalne dla czatu: Wysyłanie wiadomości prywatnych . . . . .	80
Tabela 4.4: Wymaganie funkcjonalne dla czatu: Wysyłanie wiadomości do wielu osób jednocześnie . . . . .	81
Tabela 4.5: Wymaganie funkcjonalne dla czatu: Rozpoczynanie nowego czatu . . . . .	82
Tabela 4.6: Wymaganie funkcjonalne dla czatu: Wysyłanie emotikonów .	83
Tabela 4.7: Wymaganie funkcjonalne dla czatu: Dostępność czatu po utworzeniu . . . . .	84
Tabela 4.8: Wymaganie funkcjonalne dla czatu: Edytowanie nazwy czatu grupowego . . . . .	85
Tabela 4.9: Wymaganie funkcjonalne dla czatu: Edycja zdjęcia czatu grupowego . . . . .	86
Tabela 4.10: Wymaganie funkcjonalne dla czatu: Edycja wysłanej wiadomości . . . . .	87
Tabela 4.11: Wymaganie funkcjonalne dla czatu: Dodawanie użytkowników do istniejącego czatu . . . . .	87
Tabela 4.12: Wymaganie funkcjonalne dla czatu: Wyświetlanie starszych wiadomości . . . . .	88
4.63 Profil użytkownika . . . . .	89
4.64 Lista dodanych spotów . . . . .	90
4.65 Dodanie spota . . . . .	91
4.66 Lista zdjęć . . . . .	92
4.67 Lista filmów . . . . .	92
4.68 Lista znajomych . . . . .	93
4.69 Lista obserwujących . . . . .	93
4.70 Lista obserwowanych . . . . .	94
4.71 Lista polubionych/odwiedzonych/planowanych spotów . . . . .	94
4.72 Lista komentarzy . . . . .	95
4.73 Ustawienia profilu . . . . .	96
4.74 Resetowanie hasła . . . . .	97

4.75 Dodawanie do znajomych . . . . .	98
4.76 Logowanie i rejestracja . . . . .	99
4.77 Strona główna — podstawowe filtry . . . . .	100
4.78 Strona główna — zaawansowane filtry . . . . .	101
4.79 Ustawienia motywu (ręczna zmiana) . . . . .	102
4.80 Zapamiętanie preferencji motywu . . . . .	103
4.81 Szybki przełącznik motywu w interfejsie . . . . .	104
Tabela 4.1: Wymaganie pozafunkcjonalne dla czatu: Ograniczenie widoczności czatów do członków . . . . .	106
Tabela 4.2: Wymaganie pozafunkcjonalne dla czatu: Wymóg zalogowania do korzystania z czatu . . . . .	107
Tabela 4.3: Wymaganie pozafunkcjonalne dla czatu: Grupowanie wiadomości według daty wysłania . . . . .	108
Tabela 4.4: Wymaganie pozafunkcjonalne dla czatu: Wyraźne oznaczenie nadawcy i czasu wysłania . . . . .	109
Tabela 4.5: Wymaganie pozafunkcjonalne dla czatu: Czas załadowania starszych wiadomości poniżej 3 sekund . . . . .	111
Tabela 4.6: Wymaganie pozafunkcjonalne dla czatu: Natychmiastowe wysyłanie wiadomości . . . . .	112
Tabela 4.7: Wymaganie pozafunkcjonalne dla czatu: Zachowanie wiadomości przy chwilowej utracie połączenia . . . . .	113
Tabela 4.8: Wymaganie pozafunkcjonalne dla czatu: Limit wysyłanych wiadomości w jednostce czasu . . . . .	114
Tabela 7.1: Zestawienie endpointów: panelu użytkownika . . . . .	125
Tabela 7.2: Zestawienie endpointów: modułu spotów . . . . .	126
Tabela 7.3: Zestawienie endpointów: komentarzy do spotów . . . . .	127
Tabela 7.4: Zestawienie endpointów: postów forum . . . . .	127
Tabela 7.5: Zestawienie endpointów: komentarzy forum . . . . .	128
Tabela 7.6: Zestawienie endpointów: konta użytkownika i autoryzacji . .	128
Tabela 7.7: Zestawienie endpointów: integracji GIF-ów . . . . .	129
Tabela 7.8: Zestawienie endpointów: modułu czatu . . . . .	129

Tabela 7.9: Karta endpointu: /public/user-dashboard/profile/{targetUsername}	130
Tabela 7.10: Karta endpointu: /user-dashboard/favorite-spots . . . . .	131
Tabela 7.11: Karta endpointu: /user-dashboard/photos . . . . .	132
Tabela 7.12: Karta endpointu: /user-dashboard/add-spot . . . . .	133
Tabela 7.13: Karta endpointu: /user-dashboard/add-spot . . . . .	134
Tabela 7.14: Karta endpointu: /public/spot/gallery . . . . .	135
Tabela 7.15: Karta endpointu: /public/spot/current-view . . . . .	136
Tabela 7.16: Karta endpointu: /public/spot/get-spot-basic-weather . . .	137
Tabela 7.17: Karta endpointu: /public/spot/get-spot-detailed-weather . .	138
Tabela 7.18: Karta endpointu: /public/spot/get-spot-wind-speeds . . . .	139
Tabela 7.19: Karta endpointu: /public/spot/most-popular . . . . .	140
Tabela 7.20: Karta endpointu: /public/spot/search/home-page . . . . .	141
Tabela 7.21: Karta endpointu: /public/spot/search/home-page/locations	142
Tabela 7.22: Karta endpointu: /public/spot/search/home-page/advance .	143
Tabela 7.23: Karta endpointu: /public/spot/{spotId}/comments . . . . .	145
Tabela 7.24: Karta endpointu: /public/spot/{spotId}/comments/{commentId}	146
Tabela 7.25: Karta endpointu: /spot/{spotId}/comments . . . . .	147
Tabela 7.26: Karta endpointu: /spot/comments/{commentId}/vote . . .	148
Tabela 7.27: Karta endpointu: /spot/comments/vote-type . . . . .	148
Tabela 7.28: Karta endpointu: /public/post/{postId} . . . . .	150
Tabela 7.29: Karta endpointu: /post . . . . .	151
Tabela 7.30: Karta endpointu: /post/{postId} . . . . .	152
Tabela 7.31: Karta endpointu: /post/{postId}/vote . . . . .	152
Tabela 7.32: Karta endpointu: /public/categories-tags . . . . .	153
Tabela 7.33: Karta endpointu: /public/post/{postId}/comments . . . . .	155
Tabela 7.34: Karta endpointu: /post/{postId}/comments . . . . .	156
Tabela 7.35: Karta endpointu: /post/comments/{commentId} . . . . .	157
Tabela 7.36: Karta endpointu: /post/comments/{commentId}/vote . . .	157
Tabela 7.37: Karta endpointu: /comments/{commentId}/replies . . . . .	158
Tabela 7.38: Karta endpointu: /public/account/register . . . . .	159
Tabela 7.39: Karta endpointu: /public/account/login . . . . .	159

Tabela 7.40: Karta endpointu: /public/account/forgot-password . . . . .	160
Tabela 7.41: Karta endpointu: /public/account/set-new-password . . . . .	161
Tabela 7.42: Karta endpointu: /account/check . . . . .	161
Tabela 7.43: Karta endpointu: /account/login-success . . . . .	162
Tabela 7.44: Karta endpointu: /gifs/trending . . . . .	163
Tabela 7.45: Karta endpointu: /gifs/search . . . . .	164
Tabela 7.46: Karta endpointu: /chats/{chatId}/messages . . . . .	165
Tabela 7.47: Karta endpointu: /chats/get-or-create-private-chat . . . . .	166
Tabela 7.48: Karta endpointu: /chats/{chatId}/send-files . . . . .	167
Tabela 7.49: Karta endpointu: /chats/create/group . . . . .	168
Tabela 7.50: Karta endpointu: /chats/{chatId} . . . . .	169
Tabela 7.51: Karta endpointu: /chats/group-chat/add/search/{chatId} .	170

# Bibliografia

- [1] *Disciplined Agile Delivery*. PMI. 1 stycznia 2025. URL: <https://www.pmi.org/disciplined-agile/process/introduction-to-dad/why> (dostęp 30.10.2025).
- [2] *Disciplined Agile Delivery — Lean Life Cycle*. PMI. 1 stycznia 2025. URL: <https://www.pmi.org/disciplined-agile/lifecycle/lean-lifecycle> (dostęp 30.10.2025).
- [3] Stanisław Wrycza, Bartosz Marcinkowski i Krzysztof Wyrzykowski. „Język UML 2.0 w modelowaniu systemów informatycznych”. Warszawa: Helion, 2006. ISBN: 83-736-1892-9, 8373618929.
- [4] Michał Wolski. *10 wskazówek poprawiających modelowanie procesów biznesowych w notacji BPMN*. 14 maja 2024. URL: <https://wolski.pro/2024/05/10-wskazovek-poprawiajacych-modelowanie-procesow-biznesowych-w-notacji-bpmn/> (dostęp 19.11.2025).
- [5] *About billing for GitHub Actions*. GitHub Docs. 1 stycznia 2024. URL: <https://docs.github.com/en/billing/managing-billing-for-github-actions/about-billing-for-github-actions> (dostęp 2.11.2025).
- [6] *Scalability and performance targets for Blob storage*. Microsoft Learn. 1 stycznia 2024. URL: <https://learn.microsoft.com/azure/storage/blobs/scalability-targets> (dostęp 2.11.2025).
- [7] *What are the limitations in Mailtrap?* Mailtrap Docs. 1 stycznia 2024. URL: <https://help.mailtrap.io/article/111-what-are-the-limitations-in-mailtrap/> (dostęp 2.11.2025).
- [8] *LocationIQ Pricing*. LocationIQ. 1 stycznia 2024. URL: <https://locationiq.com/pricing> (dostęp 2.11.2025).
- [9] *Google Maps (Maps URLs)*. Google Maps. 1 stycznia 2024. URL: <https://developers.google.com/maps/documentation/urls/get-started?hl=pl> (dostęp 2.11.2025).
- [10] *OpenFreeMap Documentation*. OpenFreeMap. 1 stycznia 2024. URL: <https://openfreemap.org/docs> (dostęp 2.11.2025).

- [11] *Open-Meteo API Usage & Pricing*. Open-Meteo. 1 stycznia 2024. URL: <https://open-meteo.com/en/docs/usage-and-pricing> (dostęp 2.11.2025).
- [12] *Tenor API — Documentation*. Tenor. 1 stycznia 2024. URL: <https://tenor.com/gifapi/documentation> (dostęp 2.11.2025).
- [13] *Where the ISS at? API*. wheretheiss.at. 1 stycznia 2024. URL: <https://wheretheiss.at/> (dostęp 2.11.2025).
- [14] *React useState*. 1 stycznia 2025. URL: <https://react.dev/reference/react/useState> (dostęp 3.11.2025).
- [15] *Redux*. 1 stycznia 2025. URL: <https://redux.js.org/> (dostęp 3.11.2025).
- [16] *Axios*. 1 stycznia 2025. URL: <https://axios-http.com/> (dostęp 3.11.2025).
- [17] *Tanstack Query*. 1 stycznia 2025. URL: <https://tanstack.com/query/latest> (dostęp 3.11.2025).
- [18] *Tailwind*. 1 stycznia 2025. URL: <https://tailwindcss.com/> (dostęp 3.11.2025).
- [19] *Motion*. 1 stycznia 2025. URL: <https://motion.dev/> (dostęp 3.11.2025).

# Załączniki

Płyta CD z następującą zawartością:

- *pliki projektowe* – pliki składające się na całość projektu
  - repozytorium kodu źródłowego wraz z instrukcją zbudowania i uruchomienia projektu
  - źródło pracy inżynierskiej.
- *Langmesser Adam\_Redosz Mateusz\_Oziemczuk Stanisław\_Badek Kacper\_praca pisemna* – katalog zawierający plik PDF z pracą inżynierską.