2019

Pubweb - dokumentacja

grupa

dubiński jan grębowski łukasz kaleta joanna ogonowski aleksander oniszczuk maria szymczyk kornel walkowiak paweł

Spis treści

[1. 2](#_Toc10758599)

[2. Wymagania 2](#_Toc10758600)

[2.1. Opis Metody MoSCoW 2](#_Toc10758601)

[2.2. Wymagania funkcjonalne 2](#_Toc10758602)

[2.3. Wymagania niefunkcjonalne 3](#_Toc10758603)

[3. Postanowienia ogólne 5](#_Toc10758604)

[3.1. Ogólna architektura serwisów 5](#_Toc10758605)

[4. Operacje 6](#_Toc10758606)

[4.1. Serwis autoryzacyjny 6](#_Toc10758607)

[4.2. Serwis użytkowników 6](#_Toc10758608)

[4.3. Serwis znajomych 8](#_Toc10758609)

[4.4. Serwis zgód 9](#_Toc10758610)

[4.5. Serwis pubów 9](#_Toc10758611)

[4.6. Serwis recenzji 11](#_Toc10758612)

[5. Ogólny opis rozwiązań architektonicznych 13](#_Toc10758613)

[6. 4 Views 13](#_Toc10758614)

[6.1. Widok fizyczny rozwiązania 13](#_Toc10758615)

[6.2. Widok logiczny, diagram klas 14](#_Toc10758616)

[6.3. Diagramy aktywności 15](#_Toc10758617)

[Autoryzacja operacji 15](#_Toc10758618)

[Tworzenie konta 16](#_Toc10758619)

[5.4 Diagram sekwencji 17](#_Toc10758620)

[Dodanie recenzji pubu 17](#_Toc10758621)

[Wyświetlanie strony pubu 18](#_Toc10758622)

[5.5 Diagram komponentów 19](#_Toc10758623)

[7. Diagramy związków encji 20](#_Toc10758624)

[7.1. Diagram związków encji serwisów użytkowników i autoryzacji 20](#_Toc10758625)

[7.2. Diagram związków encji serwisów pubów i recenzji 21](#_Toc10758626)

[8. Modelowanie decyzji architektonicznych w postaci MAD 2.0 21](#_Toc10758627)

[9. Analiza zgodności z RODO 23](#_Toc10758628)

[9.1. Definicje wykorzystywanych pojęć 23](#_Toc10758629)

[9.2. Prezentacja wymagań i rozwiązań 24](#_Toc10758630)

# Opis

PubWeb jest

# Wymagania

W poniższym fragmencie opisano w sposób spriorytetyzowany wymagania funkcjonalne i niefunkcjonalne systemu. Przy opisie priorytetów realizacji poszczególnych wymagań posłużono się opisaną dalej metodą MoSCoW.

## Opis Metody MoSCoW

Metoda **MoSCoW** jest techniką priorytetyzacji wykorzystywaną w analizie biznesowej i przy tworzeniu oprogramowania w celu osiągnięcia wspólnego zrozumienia pomiędzy interesariuszami co do znaczenia jakie ma dla nich dostarczenie każdego z wymagań. Wyróżnia się następujące kategorie wymagań według metody MoSCoW:

* **M** – MUST (*musi być*): Opisuje wymaganie, które *musi być* spełnione w końcowym, finalnym rozwiązaniu
* **S** – SHOULD (*powinien być*): Reprezentuje pozycję o wysokim priorytecie, która *powinna być* zawarta w rozwiązaniu, jeżeli jest to możliwe
* **C** – COULD (*może być*): Opisuje wymaganie, które jest postrzegane jako pożądane, ale niekonieczne. Zostanie ono zawarte, jeżeli pozwolą na to czas i zasoby
* **W** – WON’T (*nie będzie*): Reprezentuje wymaganie, które – za zgodą interesariuszy – *nie będzie* implementowane w danym wydaniu, ale może być rozpatrzone w przyszłości

## Wymagania funkcjonalne

**MUST:**

* system umożliwia utworzenie konta umożliwiającego korzystanie z usług systemu
* system umożliwia logowanie
* system umożliwia dodawanie recenzji pubów
* system umożliwia dodawanie nowych pubów
* użytkownik ma możliwość wyszukania w bazie pubów na podstawie nazw lub zadanych słów kluczowych, lub miasta

**SHOULD:**

* system umożliwia personalizacje profilu użytkownika
* system umożliwia komentowanie recenzji
* system umożliwia dodawanie zdjęć pubów
* system umożliwia dodanie pubu do listy do odwiedzenia
* system przedstawia rekomendacje pubów

**COULD:**

* system umożliwia dodawanie znajomych
* system zawiera system rekomendacji pubów
* system prowadzi dziennik aktywności użytkownika

**WON’T:**

* system umożliwia wyszukiwanie pubów w wybranej lokalizacja
* system wyświetla lokalizację pubów na mapie
* system umożliwia logowanie za pomocą innych platform

## Wymagania niefunkcjonalne

**MUST:**

* system opiera się na architekturze rozproszonej w postaci mikroserwisów
* struktura systemu umożliwia dodawanie nowych funkcjonalności i dalszy rozwój
* system musi zapewnić skalowalność
* system jest zgodny z RODO [Patrz: Analiza zgodności z RODO]
* wszyscy użytkownicy mogą zgłaszać błędy na dedykowany adres email
* system spełnia wymagania bezpieczeństwa [Patrz: Kwestie bezpieczeństwa]
* system implementuje protokół OAuth 2.0
* w systemie implementowane są następujące role różniące się poziomami uprawnień
  + niezalogowany użytkownik
  + użytkownik
  + administrator
* system powinien realizować strukturę zabezpieczeń w dostępie do jego zasobów w oparciu o System uprawnień nadawanych Rolom
* interfejs klienta otwiera się w przeglądarkach:
* Chrome 68 +
* Mozilla 61 +

**SHOULD:**

* Interfejs klienta jest dostosowany do urządzeń mobilnych
* System zapewnia najlepszy możliwy User Experience
* Językiem systemu jest język polski
* W aplikacja pola obowiązkowe są oznaczone w inny sposób niż pola nieobowiązkowe

**COULD:**

* w systemie implementowana jest oddzielna rola o poziomie uprawnień większym niż użytkownik dla użytkownika-właściciela pubu
* System cenzuruje słowa wulgarne

**WON’T:**

* System posiada angielską wersję językową
* System posiada dedykowaną aplikacje mobilną

# Postanowienia ogólne

## Ogólna architektura serwisów

Mikroserwisy pisane są w Javie. Interfejsy serwisów opisane są w poniższym dokumencie, dodatkowo dla serwisu autoryzacyjnego, użytkowników, zgód i znajomych wygenerowane zostały specyfikacje w formacie OPENAPI - [*https://swagger.io/specification/*](https://swagger.io/specification/) . Do uwierzytelnienia w serwisach wykorzystywane są tokeny JWT. Ich działanie opisuje dokument JWT.docx.

Wyróżniamy serwisy:

User service – serwis udostępniający operacje i informacje o użytkownikach. (Java)

Permission service – serwis udostępniający operacje i informacje o zgodach. (Java)

Friend service – serwis udostępniający informacje i operacje dotyczące relacji pomiędzy poszczególnymi użytkownikami. (Java)

Pub service – serwis udostępniający operacje i informacje o pubach. (Java)

Rating service – serwis udostępniający informacje i operacje dotyczące ocen i recenzji pubów tworzonych przez użytkowników.

# Opis funkcjonalności projektu

## Rodzaje użytkowników

## Rejestracja I logowanie

## Wyszukiwanie pubów

## Dodawanie pubu

## Zarządzanie profilem

## Wysylanie zgłoszeń

## Panel administratora

TODO

# Operacje

## Serwis autoryzacyjny

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ścieżka względna** | **Metoda HTTP** | **Znaczenie** | **Typ przyjmowany** | **Typ zwracany** | **Wymagane uprawnienia** | **Informacje dodatkowe** |
| /auth/check | GET | Zwraca informacje o tokenie JWT | {}, token w headerze Authorization | CheckAuthResponse | Brak |  |
| /auth/signin | POST | Przyjmuje login i hasło użytkownika w celu wygenerowania tokenu JWT | LoginRequest | jwtAuthenticationResponse | Brak |  |

## Serwis użytkowników

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ścieżka względna** | **Metoda HTTP** | **Znaczenie** | **Typ przyjmowany** | **Typ zwracany** | **Wymagane uprawnienia** | **Informacje**  **dodatkowe** |
| /users | GET | Pobiera listę użytkowników i ich danych publicznych | {} | [GetUserResponse], tylko informacje widoczne w kontekście JWT z zapytania | Zalogowany | Stronnicowanie |
| /users/{id} | GET | Pobiera dane konkretnego użytkownika |  | GetUserResponse, tylko informacje widoczne w kontekście JWT z zapytania | Zalogowany |  |
| /users | POST | Dodaje nowego użytkownika | SignUpRequest | GetUserResponse | Brak |  |
| /users/{id} | DELETE | Usuwa dane użytkownika o podanym id | {} | {} | Kontekst własny |  |
| /users/{id}/displaySettings | PUT | Aktualizuje dane wyświetlania użytkownika | UDisplaySettings | GetUserResponse | Kontekst własny |  |
| /users/{id}/personalInformation | PUT | Aktualizuje dane osobiste użytkownika | UserPersonalInfoDto | GetUserResponse | Kontekst własny + zgoda | Zgoda o przetwarzaniu danych osobowych |
| /users/{id}/friends | GET | Pobiera dane o znajomościach użytkownika | {} | FriendshipInfo[] | Zalogowany |  |
| /users/{id}/permission | GET | Pobiera dane o zgodach (aktywnych), które zaakceptował dany użytkownik | {} | AcceptedPermission[] | Kontekst własny |  |
| /users/{id}/avatar | GET | Pobiera avatar/zdjęcie użytkownika | {} | String z awatarem w Base64 | Zalogowany |  |
| /users/{id}/avatar | POST/PUT | Dodaje avatar/zdjęcie użytkownika | AddAvatarReqeust | {} | Kontekst własny |  |
| /users/{id}/avatar | DELETE | Usuwa avatar/zdjęcie użytkownika |  | {} | Kontekst własny |  |

## Serwis znajomych

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ścieżka względna** | **Metoda HTTP** | **Znaczenie** | **Typ przyjmowany** | **Typ zwracany** | **Wymagane uprawnienia** | **Informacje**  **Dodatkowe** |
| /requests | GET | Pobiera informacje o wszystkich zaproszeniach do znajomych | {} | Lista zaproszeń do znajomych | Kontekst własny |  |
| /requests/sent | GET | Pobiera informacje o zaproszeniach wysłanych | {} | Lista zaproszeń wysłanych | Kontekst własny |  |
| /requests/received | GET | Pobiera informacje o zaproszeniach wychodzących | {} | Lista zaproszeń otrzymanych | Kontekst własny |  |
| /requests/{id}/confirm | POST | Potwierdza dodanie do znajomych | {} | Informacje o potwierdzonym zaproszeniu | Kontekst własny | Tylko zaproszenia przychodzące do danego użytkownika |
| /requests/{id}/cancel | POST | Odrzuca zaproszenie do znajomych | {} | Informacje o odrzuconym zaproszeniu | Kontekst własny | Tylko zaproszenia przychodzące do danego użytkownika |

## Serwis zgód

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ścieżka względna** | **Metoda HTTP** | **Znaczenie** | **Typ przyjmowany** | **Typ zwracany** | **Wymagane uprawnienia** | **Informacje**  **Dodatkowe** |
| /permissions | GET | Pobiera wszystkie informacje o zgodach | {} | permissionInfo[] | Brak |  |
| /permissions/{id} | GET | Pobiera dane jednej ze zgód | {} | PermissionInfo | Brak |  |
| /permission/accepted | GET | Pobiera dane o zaakceptowanych zgodach | {} | AcceptedPermission[] | Kontekst własny |  |
| /permission/{id}/accept | POST | Akceptuje zgodę o danym id | {} | AcceptedPermission[] – lista po akceptacji | Kontekst własny |  |
| /permission/{id}/revoke | POST | Wycofuje zgodę o danym id | {} | AcceptedPermission[] – lista po wycofaniu akceptacji | Kontekst własny |  |

## Serwis pubów

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ścieżka względna** | **Metoda HTTP** | **Znaczenie** | **Typ przyjmowany** | **informacje zwrotne** | **Błędy** | **Uprawnienia** |
| /pubs | GET | lista pubow | {} | BasicIPubsInfo [] |  |  |
| /pubs/{id} | GET | info szczegolowe | {} | SpecifiicPubInfoWithTagList | IdNotFound |  |
| /pubs | POST | dodaje pub | AddPubName | PubName |  | Zalogowany |
| /pubs/{id}/info | PATCH | dodaje opis pubu | AddPubInfo | PubInfo | IdNotFound | Zalogowany |
| /pubs/{id}/info | GET | wyswietla opis | {} | PubInfo | IdNotFound |  |
| /pubs/{id}/city | PATCH | dodaje miasto | AddPubCity | PubCity | IdNotFound | Zalogowany |
| /pubs/{id}/city | GET | wyswietla miasto | {} | PubCity | IdNotFound |  |
| /pubs/{id} | DELETE | usuwa pub | {} | DeletetedPubName | IdNotFound | Admin |
| /pubs/{id}/tag | GET | Wyświetla tagi | {} | TagInfoAndId []IdNotFound | IdNotFound |  |
| /pubs/tag | GET | Puby z danym tagiem | TagName | BasicIPubsInfo [] | TagNotFound |  |
| /pubs/city | GET | Puby w mieście | CityName | BasicIPubsInfo [] | CitydNotFound |  |
| /pubs/{id}/photo | POST | Dodaje zdjecie | AddPhotoFile | AddedConfirmation | UploadError | Zalogowany |
| /pubs/{id}photo | GET | Pobiera zdjecia w formacie Base64 | {} | Base64Photo [] | IdNotFound |  |
| /pubs/check | GET | Zwraca istnienie pubu | PubId | Boolinfo |  |  |

## Serwis recenzji

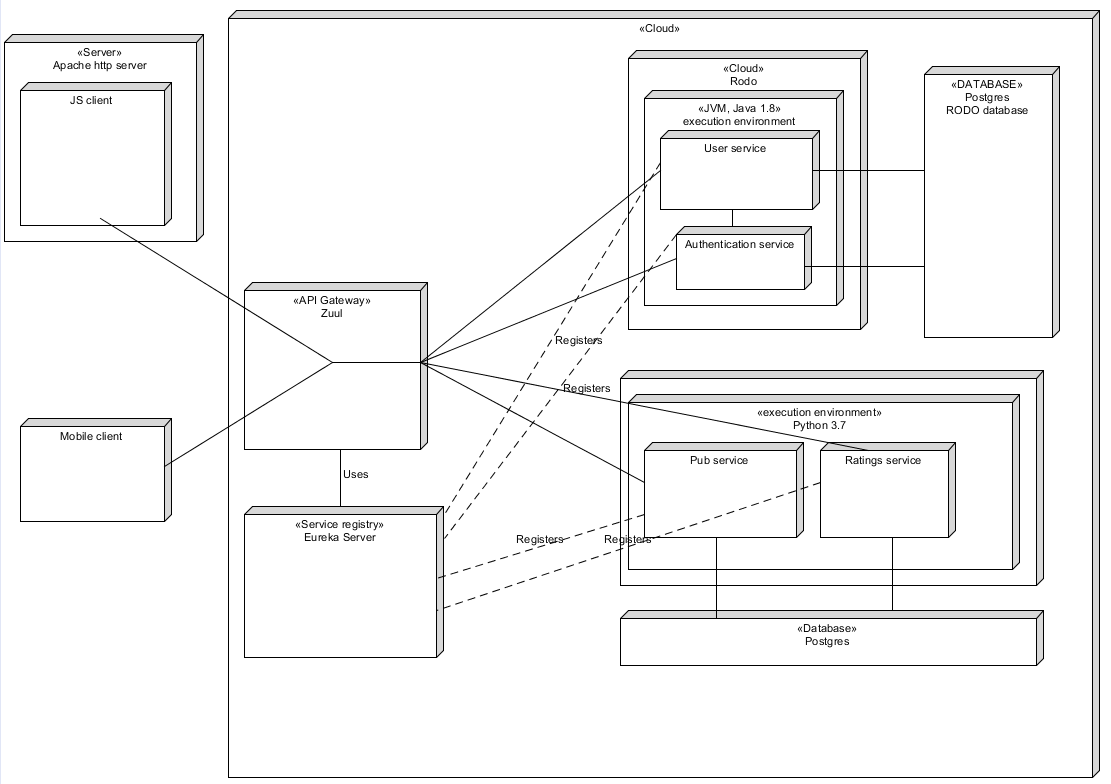
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ścieżka względna** | **Metoda HTTP** | **Działanie** | **Parametry** | **Informacje zwrotne** | **Błędy** | **Uprawnienia** |
| /reviews | GET | lista ID recenzji I ID pubów, których dotyczą | {} | IDReviewIDPub[] |  |  |
| /reviews/{id} | GET | info szczegolowe | {} | SpecificReview | IdNotFound |  |
| / reviews/{id}/opinion | PUT | dodaje tekst recenzji | AddOpinion | Opinion | IdNotFound |  |
| / reviews/{id}/opinion | GET | wyswietla tekst recenzji | {} | Opinion | IdNotFound |  |
| / reviews/{id}/stars | PUT | dodaje ocenę | AddStars | Stars | IdNotFound |  |
| / reviews /{id}/city | GET | wyświetla ocenę | {} | Stars | IdNotFound |  |
| / reviews/{id}/user\_id | PUT | dodaje ID użytkownika autora recenzji | AddUserId | UserId | IdNotFound |  |
| / reviews /{id}/user\_id | GET | wyświetla ID użytkownika autora recenzji | {} | UserId | IdNotFound |  |
| / reviews/{id}/pub\_id | PUT | dodaje ID pubu do recenzji | AddPubId | PubId | IdNotFound |  |
| / reviews /{id}/pub\_id | GET | wyświetla ID recenzowanego pubu | {} | PubId | IdNotFound |  |
| /reviews/{id} | DELETE | usuwa recenzje | {} | DeletedId | IdNotFound |  |

# Ogólny opis rozwiązań architektonicznych

W celu zrealizowania celów stawianych systemowi zdecydowaliśmy się wprowadzić architekturę opartą na mikroserwisach. Dają one możliwość konteneryzacji poszczególnych części aplikacji, a w efekcie wdrożenie w środowiskach chmurowych jest nieskomplikowane. Wzorce architektoniczne opisane są przy diagramie fizycznym.

# 4 Views

## Widok fizyczny rozwiązania



Zuul – jest to proxy ze stajni Netflixa. Zapewnia jeden wspólny interfejs dla wszystkich serwisów (realizacja wzorca *API gateway* nazywanego dalej bramą API). Pozwala ukryć implemntację i rzeczywiste rozmieszczenie serwisów. Klient aplikacji zna jedynie adres bramy API. Daje to możliwość zarówno łatwiejszego dostęp do API jak i oddzielenie interfejsów oraz implementacji. Możliwe jest również zdefiniowanie ustawień bezpieczeństwa (w omawianym przypadku niewykorzystywane, zarządzanie nimi odbywa się na poziomie aplikacji).

Eureka – jest to serwer *service discovery*. W przypadku mikroserwisów ręczne zmiany konfiguracji w momencie pojawienia się nowej instacji lub przeniesienia instniejącej instancji mikroserwisu byłyby żmudne. Eureka jest rejestrem serwisów, który wystawia API REST, dzięki czemu mikroserwis sam może zarejestrować swoją obecność pod wskazanym adresem (np. przy uruchomieniu).

Współpraca Zuul i Eureka:

Ze względu na wspólnego dostawcę Zuul i Eureka posiadają mechanizm integracji. Zuul odczytuje rejestr Eureki i trasuje zapytania do poszczególnych serwisów.

## Widok logiczny, diagram klas

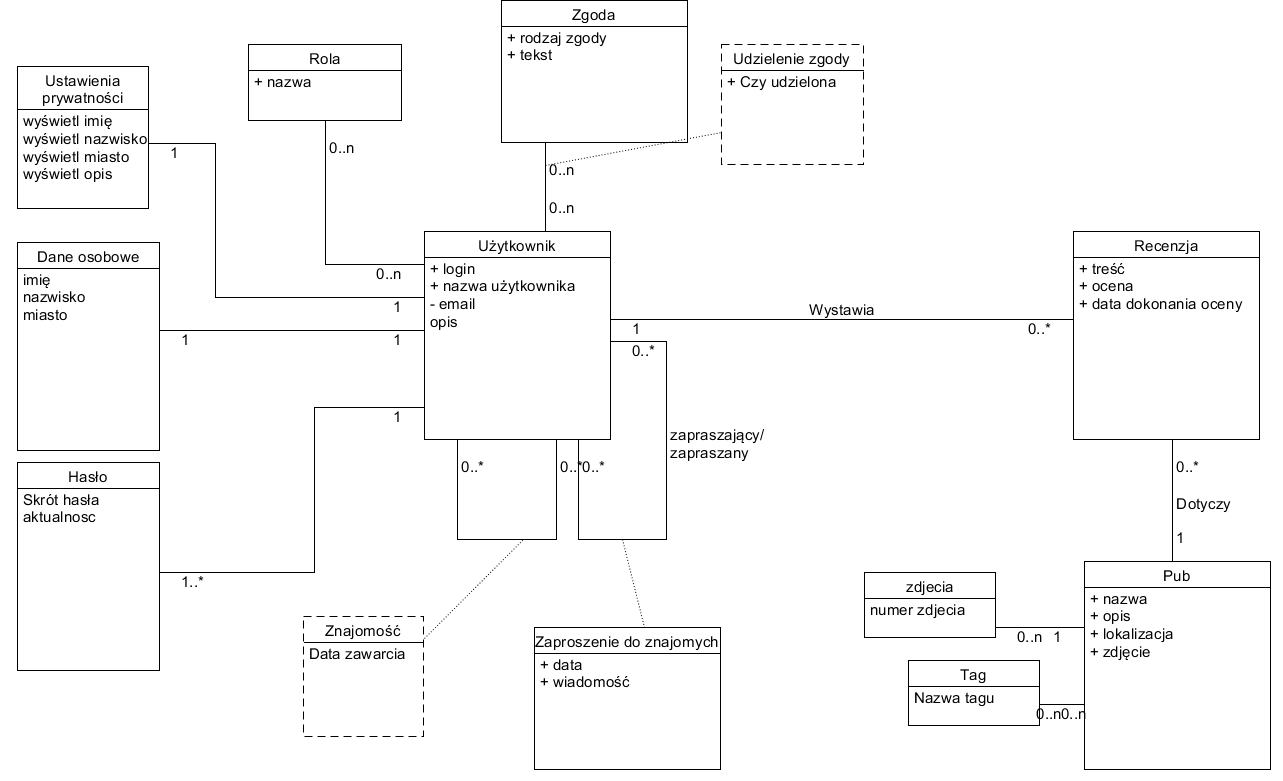
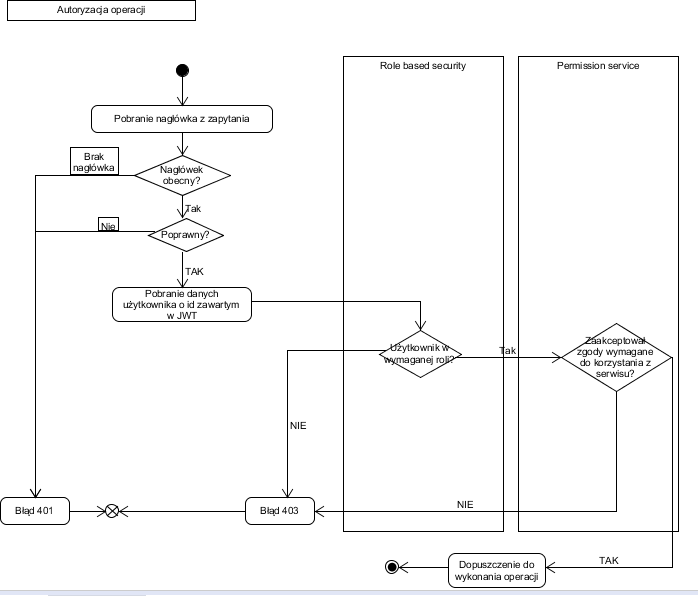


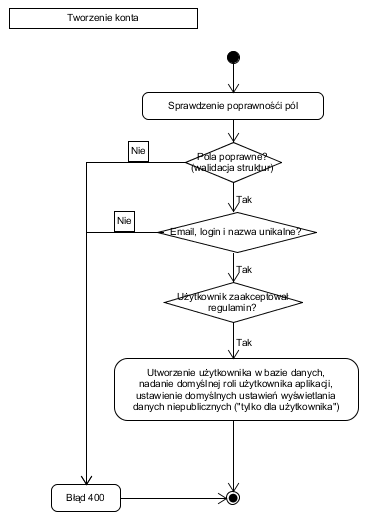
Diagram klas obrazuje logiczne zależności pomiędzy poszczególnymi komponentami systemu.

## Diagramy aktywności

### Autoryzacja operacji

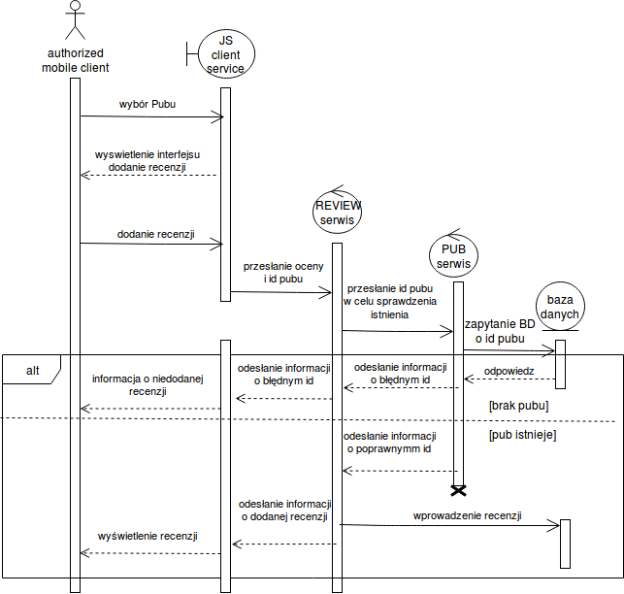


### Tworzenie konta

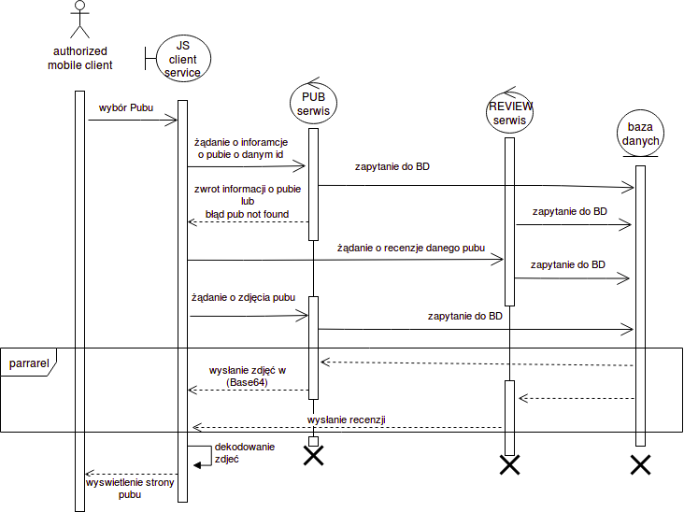


## Diagram sekwencji

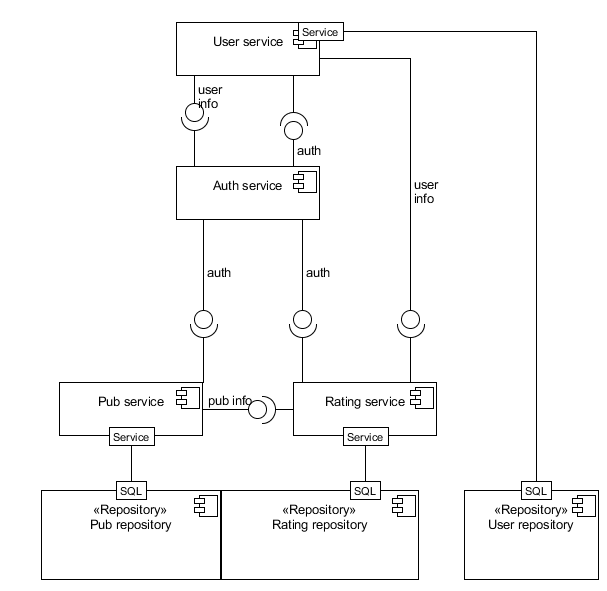
### Dodanie recenzji pubu



### Wyświetlanie strony pubu

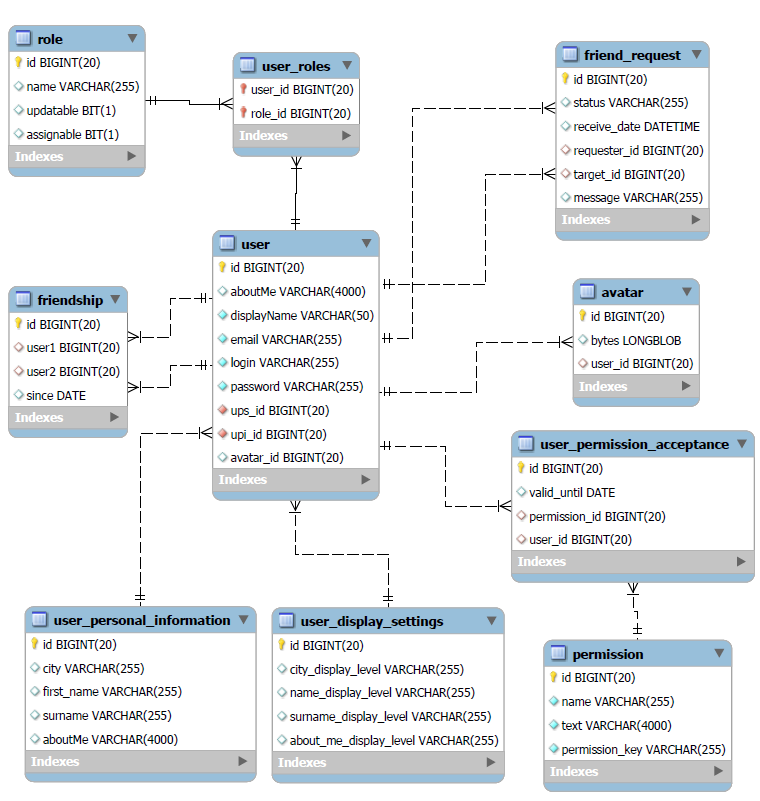


## Diagram komponentów

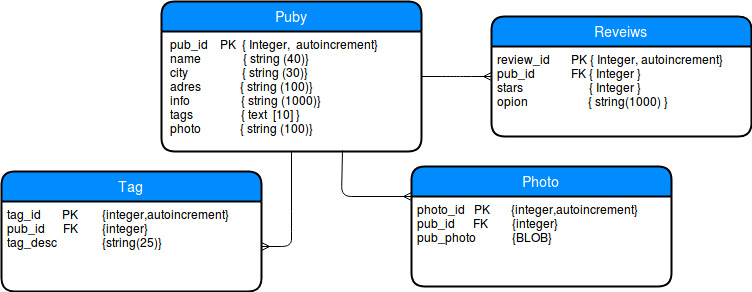


# Diagramy związków encji

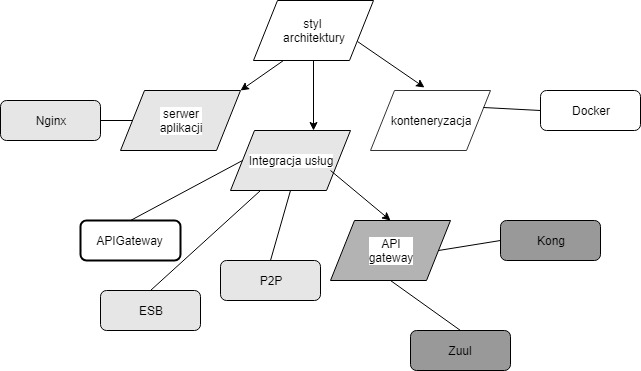
## Diagram związków encji serwisów użytkowników i autoryzacji

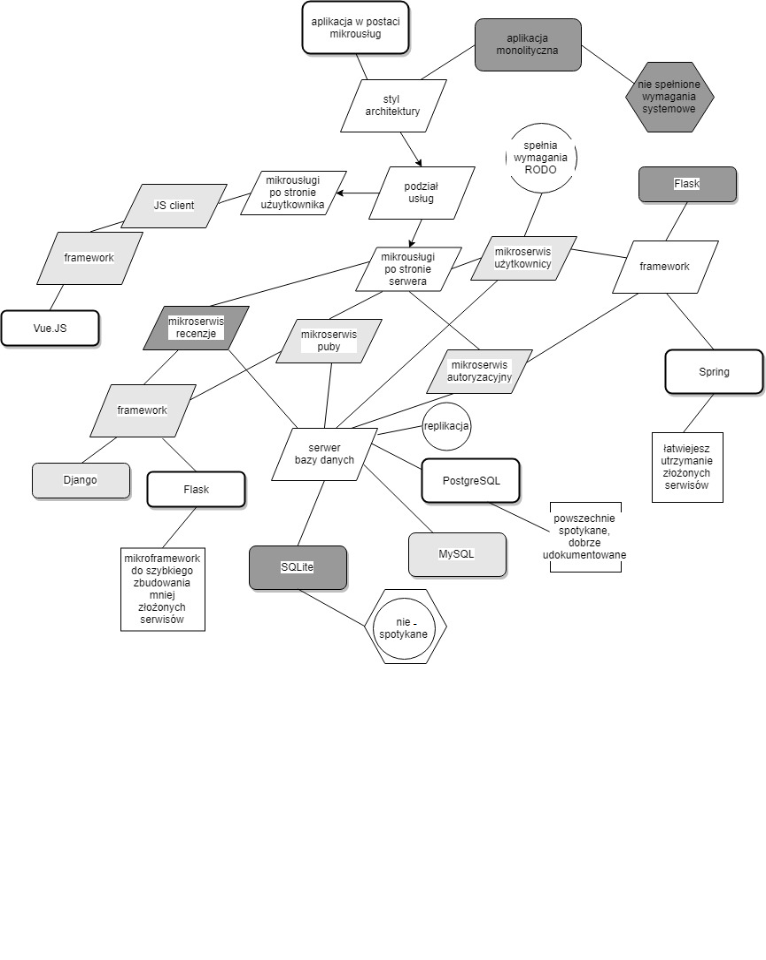


## Diagram związków encji serwisów pubów i recenzji



# Modelowanie decyzji architektonicznych w postaci MAD 2.0





# Analiza zgodności z RODO

RODO jest skrótem od Rozporządzenia o Ochronie Danych Osobowych (ang*. General Data Protection Regulation* - GDPR). Jest to rozporządzenie Parlamentu Europejskiego i Rady (UE) 2016/679 z dnia 27 kwietnia 2016 r. w sprawie ochrony osób fizycznych w związku z przetwarzaniem danych osobowych i w sprawie swobodnego przepływu takich danych oraz uchylenia dyrektywy 95/46/WE (ogólne rozporządzenie o ochronie danych). Weszło ono w życie 25 maja 2018 r.

## Definicje wykorzystywanych pojęć

Podczas analizy zgodności systemu z RODO konieczna jest znajomość wykorzystywanych w ustawie pojęć. Jest ona niezbędna do poprawnego zrozumienia wymagań stawianych systemowi wynikających z tego rozporządzenia. W tym celu

***„dane osobowe”*** oznaczają informacje o zidentyfikowanej lub możliwej do zidentyfikowania osobie fizycznej („osobie, której dane dotyczą”); możliwa do zidentyfikowania osoba fizyczna to osoba, którą można bezpośrednio lub pośrednio zidentyfikować, w szczególności na podstawie identyfikatora takiego jak imię i nazwisko, numer identyfikacyjny, dane o lokalizacji, identyfikator internetowy lub jeden bądź kilka szczególnych czynników określających fizyczną, fizjologiczną, genetyczną, psychiczną, ekonomiczną, kulturową lub społeczną tożsamość osoby fizycznej

„***przetwarzanie***” oznacza operację lub zestaw operacji wykonywanych na danych osobowych lub zestawach danych osobowych w sposób zautomatyzowany lub niezautomatyzowany, taką jak zbieranie, utrwalanie, organizowanie, porządkowanie, przechowywanie, adaptowanie lub modyfikowanie, pobieranie, przeglądanie, wykorzystywanie, ujawnianie poprzez przesłanie, rozpowszechnianie lub innego rodzaju udostępnianie, dopasowywanie lub łączenie, ograniczanie, usuwanie lub niszczenie

**„ograniczenie przetwarzania”** oznacza oznaczenie przechowywanych danych osobowych w celu ograniczenia ich przyszłego przetwarzania

**„profilowanie”** oznacza dowolną formę zautomatyzowanego przetwarzania danych osobowych, które polega na wykorzystaniu danych osobowych do oceny niektórych czynników osobowych osoby fizycznej, w szczególności do analizy lub prognozy aspektów dotyczących efektów pracy tej osoby fizycznej, jej sytuacji ekonomicznej, zdrowia, osobistych preferencji, zainteresowań, wiarygodności, zachowania, lokalizacji lub przemieszczania się;

**„pseudonimizacja”** oznacza przetworzenie danych osobowych w taki sposób, by nie można ich było już przypisać konkretnej osobie, której dane dotyczą, bez użycia dodatkowych informacji, pod warunkiem że takie dodatkowe informacje są przechowywane osobno i są objęte środkami technicznymi i organizacyjnymi uniemożliwiającymi ich przypisanie zidentyfikowanej lub możliwej do zidentyfikowania osobie fizycznej

**„zgoda**” osoby, której dane dotyczą oznacza dobrowolne, konkretne, świadome i jednoznaczne okazanie woli, którym osoba, której dane dotyczą, w formie oświadczenia lub wyraźnego działania potwierdzającego, przyzwala na przetwarzanie dotyczących jej danych osobowych

**„naruszenie ochrony danych osobowych”** oznacza naruszenie bezpieczeństwa prowadzące do przypadkowego lub niezgodnego z prawem zniszczenia, utracenia, zmodyfikowania, nieuprawnionego ujawnienia lub nieuprawnionego dostępu do danych osobowych przesyłanych, przechowywanych lub w inny sposób przetwarzanych

## Prezentacja wymagań i rozwiązań

W tej części zawarto kolejny wymagania stawiane przed systemem, które wynikają z obowiązywania RODO oraz przyjęte rozwiązania odnoszące się do tych wymagań.

**Wymaganie 1: Privacy by design**

Zgodnie z Art. 25 ust.1:

*Uwzględniając stan wiedzy technicznej, koszt wdrażania oraz charakter, zakres, kontekst i cele przetwarzania oraz ryzyko naruszenia praw lub wolności osób fizycznych o różnym prawdopodobieństwie wystąpienia i wadze zagrożenia wynikające z przetwarzania, administrator – zarówno przy określaniu sposobów przetwarzania, jak i w czasie samego przetwarzania – wdraża odpowiednie środki techniczne i organizacyjne, takie jak pseudonimizacja, zaprojektowane w celu skutecznej realizacji zasad ochrony danych, takich jak minimalizacja danych, oraz w celu nadania przetwarzaniu niezbędnych zabezpieczeń, tak by spełnić wymogi niniejszego rozporządzenia oraz chronić prawa osób, których dane dotyczą.*

**Wymaganie 2: Privacy by default**

Zgodnie z Art. 25 ust.2:

*Administrator wdraża odpowiednie środki techniczne i organizacyjne, aby domyślnie przetwarzane były wyłącznie te dane osobowe, które są niezbędne dla osiągnięcia każdego konkretnego celu przetwarzania. Obowiązek ten odnosi się do ilości zbieranych danych osobowych, zakresu ich przetwarzania, okresu ich przechowywania oraz ich dostępności. W szczególności środki te zapewniają, by domyślnie dane osobowe nie były udostępniane bez interwencji danej osoby nieokreślonej liczbie osób fizycznych*

**Realizacja 1+2:** System nie zbiera na temat użytkowników danych nie mających zastosowania przy jego działaniu. Zminimalizowana jest ilość danych wymaganych przy tworzeniu konta użytkownika w serwisie. Podanie dodatkowe informacje dotyczących użytkownika jest możliwe poprzez edycje profilu użytkownika i jest całkowicie dobrowolne. Dane użytkownika wykorzystywane są przez system rekomendacji, który przedstawia użytkownikowi spersonalizowane propozycje pubów do odpowiedzenia, jeżeli użytkownik wyrazi na to zgodę. 0 W bazie danych zawierającej dane osobowe użytkowników zastosowany jest mechanizm pseudonimizacji.

**Wymaganie 3: Brak możliwości przetwarzania, zbierania i przechowywania danych osobowych bez jednej z podstawy wskazanych w przepisach**

Zgodnie z Art. 6. Podstawy do przetwarzania danych to:

* umowa - przetwarzanie jest niezbędne do wykonania umowy
* zgoda wyrażona przez osobę, której dane dotyczą
* wypełnienie obowiązku prawnego
* realizacja prawnie uzasadnionych interesów administratora danych

**Realizacja 3:** Podstawą przetwarzania danych w systemie jest zgoda użytkownika. System przetwarza dane osobowe użytkownika jedynie w wypadku wyrażenia zgody przez użytkownika. Zgoda ta uzyskana jest w sposób prawidłowy, co opisuje *Wymaganie 4*.

**Wymaganie 4: Pozyskanie zgody od użytkownika w sposób prawidłowy**

Zgodnie z Art.7:

*1. Jeżeli przetwarzanie odbywa się na podstawie zgody, administrator musi być w stanie wykazać, że osoba, której dane dotyczą, wyraziła zgodę na przetwarzanie swoich danych osobowych.*

*2. Jeżeli osoba, której dane dotyczą, wyrażą zgodę w pisemnym oświadczeniu, które dotyczy także innych kwestii, zapytanie o zgodę musi zostać przedstawione w sposób pozwalający wyraźnie odróżnić je od pozostałych kwestii, w zrozumiałej i łatwo dostępnej formie, jasnym i prostym językiem. Część takiego oświadczenia osoby, której dane dotyczą, stanowiąca naruszenie niniejszego rozporządzenia nie jest wiążąca.*

*3. Osoba, której dane dotyczą, ma prawo w dowolnym momencie wycofać zgodę. Wycofanie zgody nie wpływa na zgodność z prawem przetwarzania, którego dokonano na podstawie zgody przed jej wycofaniem. Osoba, której dane dotyczą, jest o tym informowana, zanim wyrazi zgodę. Wycofanie zgody musi być równie łatwe jak jej wyrażenie.*

*4. Oceniając, czy zgodę wyrażono dobrowolnie, w jak największym stopniu uwzględnia się, czy między innymi od zgody na przetwarzanie danych nie jest uzależnione wykonanie umowy, w tym świadczenie usługi, jeśli przetwarzanie danych osobowych nie jest niezbędne do wykonania tej umowy*

Oznacz to, że:

• zgoda pozyskana musi być zawsze przed rozpoczęciem przetwarzania danych

• sposoby wyrażania zgody: pisemna, ustna; klarowne, potwierdzające czynność działanie

• automatyczne zaznaczanie zgody jest zabronione

• zgoda musi dotyczyć przetwarzania konkretnych danych, w konkretnym czasie, w konkretnym celu

• obowiązuje zakaz łączenia zgód na przetwarzanie danych

• zgoda musi być wyrażona bez przymusu

• zgoda musi być poparta rzetelną klauzulą informacyjną

• nie jest wyrażeniem zgody: milczenie, okienka domyślnie zaznaczone, niepodjęcie działania itp.

**Rozwiązanie 4:** Zgoda użytkownika na przetwarzanie jego danych osobowych uzyskiwana jest podczas jego rejestracji w systemie. System wyświetla użytkownikowi zapytanie o zgodę na wykorzystanie jego danych osobowych. Zapytanie to spełnia wymienione powyżej warunki. Rezygnacja ze zgodny przebiega w prosty sposób i jest możliwa poprzez jej odznaczenie w profilu użytkownika po zalogowaniu.

**Wymaganie 5: Prawo użytkownika do przeniesienia danych**

RODO wprowadza prawo osoby, której dane dotyczą do przeniesienia jej danych osobowych do innego administratora. Osoba, której dane są przetwarzane może poprosić o przekazanie danych wskazanemu przez nią podmiotowi. Dane należy przekazać w ustrukturyzowanym, powszechnie używanym formacie nadającym się do odczytu maszynowego (np. XML czy CSV).

**Rozwiązanie 5:** W aplikacji webowej podane są dane kontaktowe do administracji serwisu. Prośbę o przeniesienie danych do innego administratora należy wysłać na adres mailowy wskazany w danych kontaktowych. W treści wiadomości należy umieści token (kod), który uwierzytelni użytkownika. Token ten można wygenerować w profilu użytkownika po zalogowaniu. Mechanizm ten służy zwiększeniu bezpieczeństwa.

**Wymaganie 6: Prawo użytkownika do bycia zapomnianym**

Prawo do bycia zapomnianym to prawo do usunięcia danych. Osoba, której dotyczą dane ma prawo zażądać od administratora danych usunięcia dotyczących jej danych, a żądanie to musi być spełnione jeśli nie występują przesłanki, które to uniemożliwiają.

Bez wątpienia należy usunąć dane m.in. jeśli:

* nie są już one niezbędne do celów, w których zostały zebrane
* zgoda na podstawie, której przetwarzałeś dane została cofnięta
* dane były przetwarzane niezgodnie z prawem
* dane osobowe zostały zebrane w związku z oferowaniem usług społeczeństwa informacyjnego (np. w ramach portalu społecznościowego)

Są jednak sytuacje, w których możliwa jest odmowa spełnienia żądania usunięcia danych. Ma to miejsce m.in. jeśli przetwarzanie tych danych jest niezbędne:

* do wywiązania się z prawnego obowiązku
* do ustalenia, dochodzenia lub obrony roszczeń
* do korzystania z prawa do wolności wypowiedzi i informacji

**Rozwiązanie 6:** W aplikacji webowej podane są dane kontaktowe do administracji serwisu. Prośbę o usunięcie danych należy wysłać na adres mailowy wskazany w danych kontaktowych. W treści wiadomości należy umieści token (kod), który uwierzytelni użytkownika. Token ten można wygenerować w profilu użytkownika po zalogowaniu. Mechanizm ten służy zwiększeniu bezpieczeństwa.

**Wymaganie 7: Prawo użytkownika do żądania ograniczenia przetwarzania danych**

**Wymaganie 8: Prawo użytkownika do otrzymania kopii danych**

Rozwiązanie 8: Wszelkie dane o użytkowniku przechowywane przez aplikacje dostępne są z poziomu interfejsów aplikacji (wyłącznie dla użytkownika). Ponadto istnieje możliwość złożenia żądania wydania kopii danych przez formularz kontaktu z administracją serwisu. Kopia danych przesyłana jest na adres mailowy użytkownika.

**Wymaganie 9: Prawo użytkownika do sprzeciwu wobec profilowania**

Zgodnie z art. 4 pkt 4  RODO, profilowanie to:

 “dowolna forma zautomatyzowanego przetwarzania danych osobowych, które polega na wykorzystaniu danych osobowych do oceny niektórych czynników osobowych osoby fizycznej, w szczególności do analizy lub prognozy aspektów dotyczących efektów pracy tej osoby fizycznej, jej sytuacji ekonomicznej, zdrowia, osobistych preferencji, zainteresowań, wiarygodności, zachowania, lokalizacji lub przemieszczania się”.

**Rozwiązanie 9:** Aplikacja nie wyświetla treści personalizowanych w oparciu o dane osobowe (nie uwzględnia miasta pochodzenia). Aplikacja w wersji obecnej nie wyświetla reklam, ani sugestii. Profilowanie użytkownika nie zachodzi.

**Wymaganie 10: Przejrzystość języka**

Zabronione jest używanie trudnych, specjalistycznych sformułowań, tak samo jak utrudnianie zapoznania się z ważnymi informacjami, np. stosowania w komunikatach tzw. metody „drobnego druczku”.

**Rozwiązanie 10:** Tekst zgód napisany jest w przystępnej i zrozumiałej formie. Przejrzystość działania serwisu jest częścią jednego z wymagań niefunkcjonalnych serwisu = „zapewnienie możliwie najlepszego User Experience.