

Dokumentacja

System Zarządzania Otwartymi Danymi Publicznymi

Specyfikacja wymagań

Informacje podstawowe

Autorzy

- Grochowski Albert
- Biwejnis Oskar
- Minich Marharyta
- Rohoutseu Ilya
- Rusinovich Ivan
- Yarmats Stanislau
- Horobets Kiril
- Sawicki Grzegorz
- Marszewski Marcin
- Jeleniewicz Wiktoria
- Piluk Aleksander
- Dobkowski Kamil

Abstrakt

Dokument zawiera specyfikację wymagań dla systemu Zarządzania Otwartymi Danymi Publicznymi składającą się z wizji systemu, wymagań użytkownika oraz wymagań oprogramowania dla pierwszych iteracji projektu.

Historia zmian

Wersja	Data	Kto	Opis
1.0.0	01.03.2025	Albert Grochowski	Wersja startowa
1.1.0	07.03.2025	Oskar Biwejnis, Minich Marharyta	Dodano procesy biznesowe
1.2.0	14.03.2025	Rohoutseu Ilya, Rusinovich Ivan, Yarmats Stanislau	Dodano wizje systemu
1.3.0	21.03.2025	Horobets Kiril, Grzegorz Sawicki, Marszewski Marcin	Dodano wymagania funkcjonalne
1.4.0	28.03.2025	Jeleniewicz Wiktoria	Dodano słownik dziedziny
1.5.0	04.04.2025	Horobets Kiril, Grzegorz Sawicki, Marszewski Marcin	Dodano wymagania jakościowe

1. Wprowadzenie

1.1 Cel dokumentu

Zdefiniowanie jasnych i precyzyjnych wymagań funkcjonalnych oraz нефункциональных, które mają być spełnione przez projektowany system. Specyfikacja ta ma na celu zapewnienie, że wszystkie potrzeby użytkowników, interesariuszy oraz wymagania techniczne i operacyjne zostaną w pełni uwzględnione w procesie projektowania, implementacji i testowania systemu.

1.2 Streszczenie dla kierownictwa

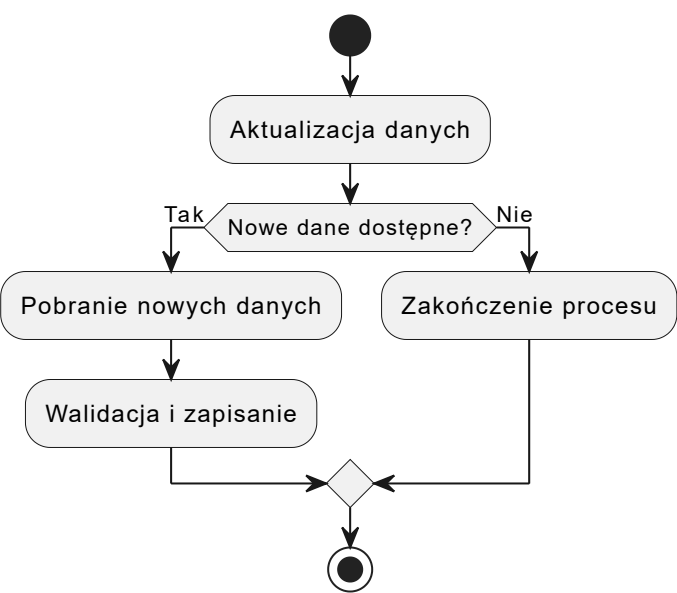
Celem projektu jest stworzenie systemu informatycznego do zarządzania publicznymi danymi otwartymi, który umożliwi sprawne udostępnianie i aktualizację danych publicznych w sposób dynamiczny, zgodny z obowiązującymi standardami krajowymi i europejskimi. System będzie wykorzystywany przez dużą jednostkę samorządową, która chce udostępniać szereg danych takich jak rozkłady jazdy komunikacji publicznej, dane o położeniu pojazdów, stężenia zanieczyszczeń w powietrzu, dostępność toalet miejskich czy godziny pracy urzędów.

2. Opis biznesu

2.1 Procesy biznesowe

(PB0001) Regularna aktualizacja danych

Diagram



Opis

Na podstawie diagramu PlantUML, można opisać proces biznesowy związany z **regularną aktualizacją danych**. Proces ten składa się z czterech głównych etapów, w których uczestniczy jedna rola:

- 1. **Sprawdzenie dostępności nowych danych**

- **Rola:** System
- **Opis:** System weryfikuje, czy nowe dane są dostępne do pobrania z odpowiedniego źródła.

2. Pobranie nowych danych

- **Rola:** System
- **Opis:** Jeśli nowe dane są dostępne, system pobiera je z odpowiedniego źródła. Może to obejmować połączenie z zewnętrznymi bazami danych lub serwisami, aby uzyskać aktualne informacje.

3. Walidacja i zapisanie danych

- **Rola:** System
- **Opis:** System sprawdza poprawność nowych danych, aby upewnić się, że są one zgodne z wymaganiami jakościowymi. Po pomyślnym zwalidowaniu, dane są zapisywane w bazie danych, co zapewnia ich aktualność.

4. Zakończenie procesu

- **Rola:** System
- **Opis:** Jeśli nie ma nowych danych do pobrania, proces kończy się. System nie przeprowadza żadnych dalszych działań.

Proces biznesowy polega na regularnym monitorowaniu dostępności nowych danych oraz ich pobieraniu i aktualizacji w systemie. Celem tego procesu jest utrzymanie bazy danych w stanie aktualności i zgodności z rzeczywistością.

Kod diagramu

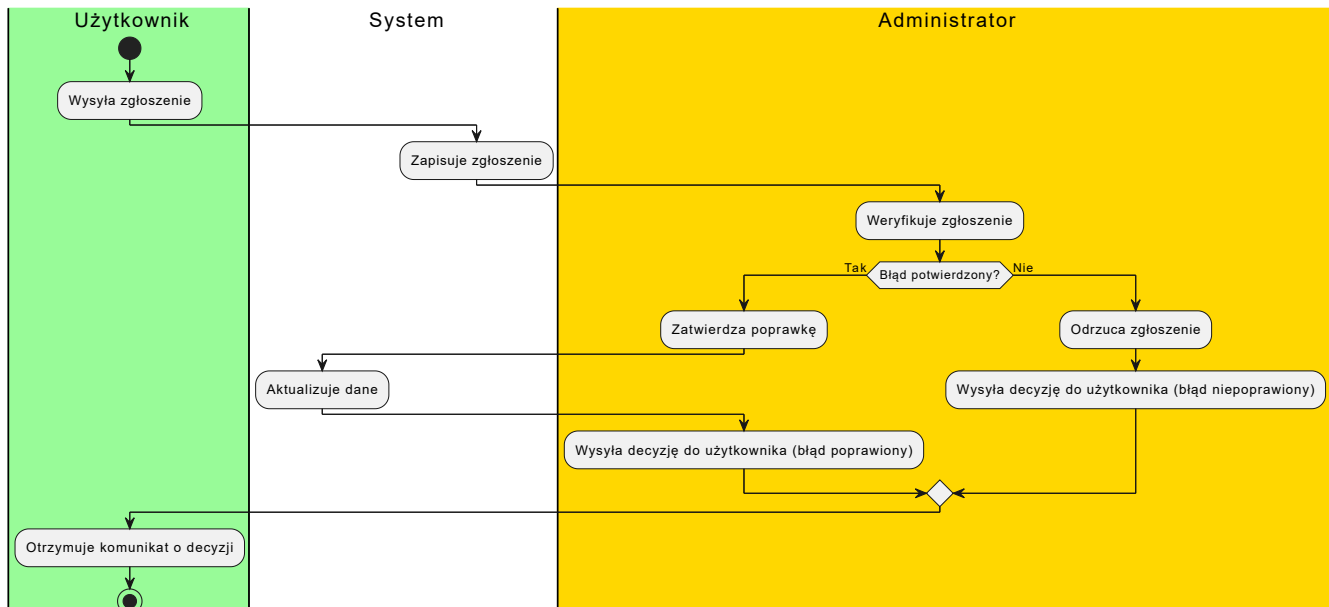
```
@startuml
start

:Aktualizacja danych;
if (Nowe dane dostępne?) then (Tak)
  :Pobranie nowych danych;
  :Walidacja i zapisanie;
else (Nie)
  :Zakończenie procesu;
endif

stop
@enduml
```

(PB0002) Zgłaszanie błędów danych przez użytkowników

Diagram



Opis

Na podstawie diagramu PlantUML, można opisać proces biznesowy związany z **zgłaszaniem błędów danych przez użytkowników**. Proces ten składa się z kilku głównych etapów, w których uczestniczą trzy role:

1. Zgłoszenie błędu

- Rola: Użytkownik
- Opis: Użytkownik wysła zgłoszenie o błędzie w danych, które mogą dotyczyć niepoprawnych, niekompletnych lub błędnych informacji znajdujących się w systemie.

- Kolor: #palegreen
- Oznaczenie: u

2. Zapisanie zgłoszenia

- Rola: System
- Opis: System rejestruje zgłoszenie w bazie danych, co pozwala na dalsze śledzenie zgłoszonego problemu i jego rozwiązanie.
- Oznaczenie: s

3. Weryfikacja przez administratora

- Rola: Administrator
- Opis: Administrator sprawdza zgłoszenie, analizując jego zasadność i określając, czy rzeczywiście występuje błąd w danych. Administrator podejmuje decyzję na podstawie dostępnych informacji.
- Kolor: #gold
- Oznaczenie: a

4. Decyzja

- Rola: Administrator
- Opis: Administrator podejmuje decyzję:
 - Jeśli błąd jest potwierdzony, administrator zatwierdza poprawkę.
 - Jeśli błąd nie jest potwierdzony, administrator odrzuca zgłoszenie.

- Kolor: #gold
- Oznaczenie: a

5. Aktualizacja danych (jeśli błąd został potwierdzony)

- Rola: System
- Opis: System aktualizuje dane, aby były poprawne.
- Oznaczenie: s

6. Informacja zwrotna dla użytkownika

- Rola: Użytkownik
- Opis: Użytkownik otrzymuje komunikat o decyzji administratora dotyczącej zgłoszonego błędu.
- Kolor: #palegreen
- Oznaczenie: u

Proces biznesowy polega na umożliwieniu użytkownikom zgłaszania błędów w danych, co pozwala na utrzymanie jakości informacji w systemie. Administratorzy mają za zadanie zweryfikować zgłoszenia i podjąć odpowiednie działania, aby zapewnić poprawność danych.

Kod diagramu

```
@startuml
|#palegreen|u| Użytkownik
start
:Wysyła zgłoszenie;

|s| System
:Zapisuje zgłoszenie;

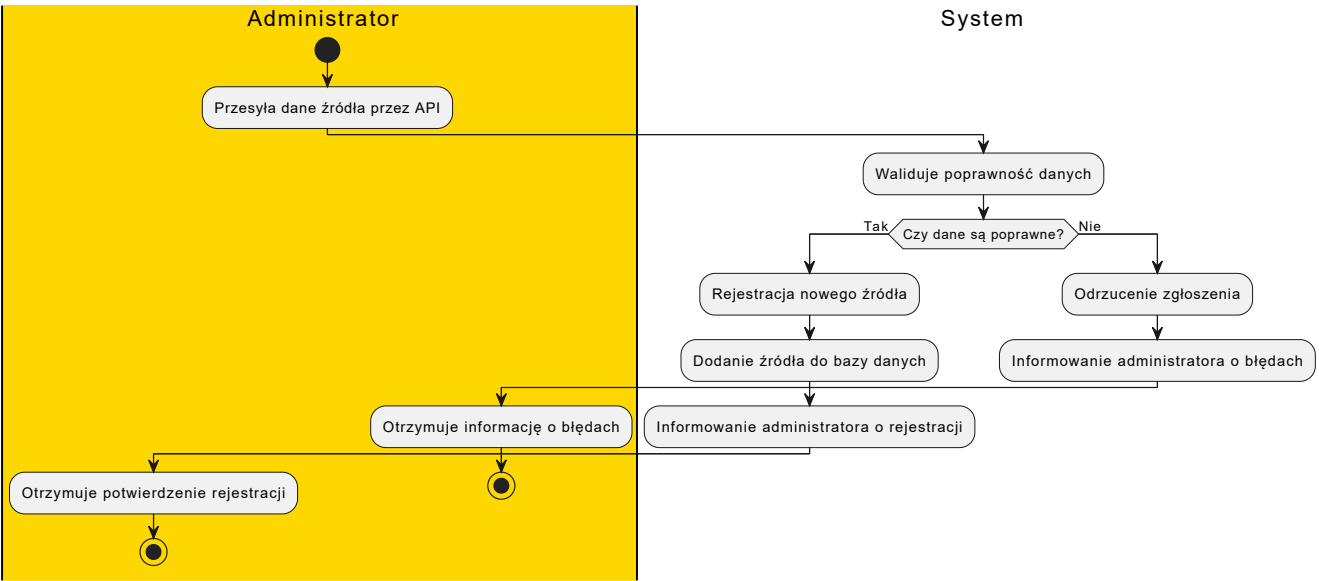
|#gold|a| Administrator
:Weryfikuje zgłoszenie;

if (Błąd potwierdzony?) then (Tak)
:Zatwierdza poprawkę;
|s| System
:Aktualizuje dane;
|#gold|a| Administrator
:Wysyła decyzję do użytkownika (błąd poprawiony);
else (Nie)
:Odrzuca zgłoszenie;
:Wysyła decyzję do użytkownika (błąd niepoprawiony);
endif

|#palegreen|u| Użytkownik
:Otrzymuje komunikat o decyzji;
stop
@enduml
```

(PB0003) Rejestrowanie nowych źródeł danych poprzez api

Diagram



Opis

Na podstawie diagramu PlantUML, można opisać proces biznesowy związany z **rejestrowaniem nowych źródeł danych poprzez API**. Proces ten składa się z trzech głównych etapów, w których uczestniczą trzy różne role:

1. Przesyłanie danych przez administratora (Send Data by Administrator):

- Rola: Administrator (administrator)
- Opis: Administrator przesyła dane nowego źródła do systemu poprzez API. Dane te mogą obejmować różne informacje o źródle, takie jak adres URL, parametry dostępu i inne niezbędne informacje.
- Kolor: #gold
- Oznaczenie: a

2. Walidacja danych (Data Validation):

- Rola: System (system)
- Opis: System sprawdza poprawność przesłanych danych, weryfikując, czy spełniają one określone wymagania, takie jak format, kompletność i zgodność z innymi danymi w systemie.
- Oznaczenie: s

3. Rejestracja nowego źródła (Register New Source):

- Rola: System (system)
- Opis: Jeśli dane są poprawne, nowe źródło zostaje zapisane w bazie danych. Jeśli walidacja nie powiedzie się, zgłoszenie zostaje odrzucone, a administrator zostaje poinformowany o błędach.
- Oznaczenie: s

Proces biznesowy umożliwia administratorowi rejestrowanie nowych źródeł danych w systemie w sposób zautomatyzowany i bezpieczny. Dzięki API proces rejestracji jest szybki i efektywny, a walidacja danych zapewnia, że tylko poprawne informacje trafiają do systemu, co poprawia jakość danych i bezpieczeństwo operacji.

Kod diagramu

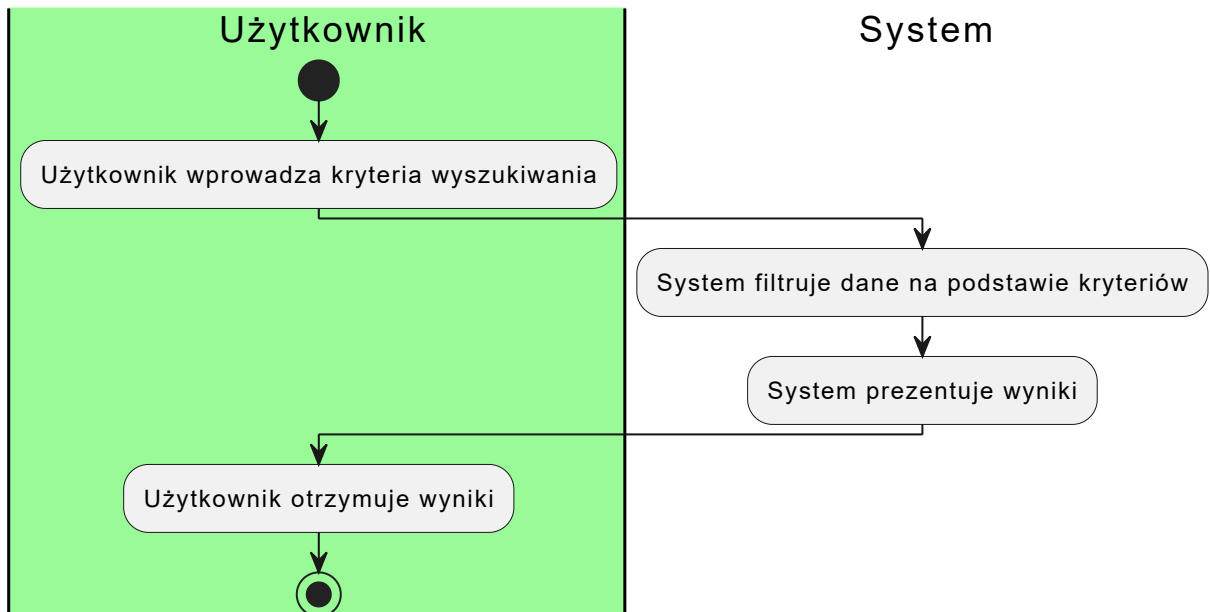
```
@startuml
|#gold|a| Administrator
start
:Przesyła dane źródła przez API;

|s| System
:Waliduje poprawność danych;

if (Czy dane są poprawne?) then (Tak)
:Rejestracja nowego źródła;
:Dodanie źródła do bazy danych;
:Informowanie administratora o rejestracji;
|#gold|a| Administrator
:Otrzymuje potwierdzenie rejestracji;
stop
else (Nie)
|s| System
:Odrzucenie zgłoszenia;
:Informowanie administratora o błędach;
|#gold|a| Administrator
:Otrzymuje informację o błędach;
stop
endif
@enduml
```

(PB0004) Filtrowanie wyszukiwanie danych

Diagram



Opis

Na podstawie diagramu PlantUML, można opisać proces biznesowy związany z **filtrowaniem wyszukiwania danych**:

1. Podanie kryteriów wyszukiwania

- Rola: Użytkownik (user)
- Opis: Użytkownik wprowadza filtry i kryteria wyszukiwania, które pozwolą na określenie, jakie dane mają zostać znalezione.
- Kolor: #palegreen
- Oznaczenie: **u**

2. Filtrowanie danych

- Rola: System (system)
- Opis: System przetwarza zapytanie użytkownika, przeszukując dostępne dane i stosując odpowiednie filtry, by znaleźć pasujące wyniki.
- Oznaczenie: **s**

3. Prezentacja wyników

- Rola: System (system)
- Opis: Po przetworzeniu zapytania system wyświetla użytkownikowi wyniki wyszukiwania w odpowiednim formacie, który może być dostosowany do potrzeb użytkownika (np. tabela, lista, grafika).
- Oznaczenie: **s**

Proces biznesowy polega na zapewnieniu użytkownikowi łatwego dostępu do precyzyjnych informacji w oparciu o zadane kryteria. Wspomaga to podejmowanie decyzji, umożliwiając szybkie filtrowanie i przetwarzanie dużych zbiorów danych.

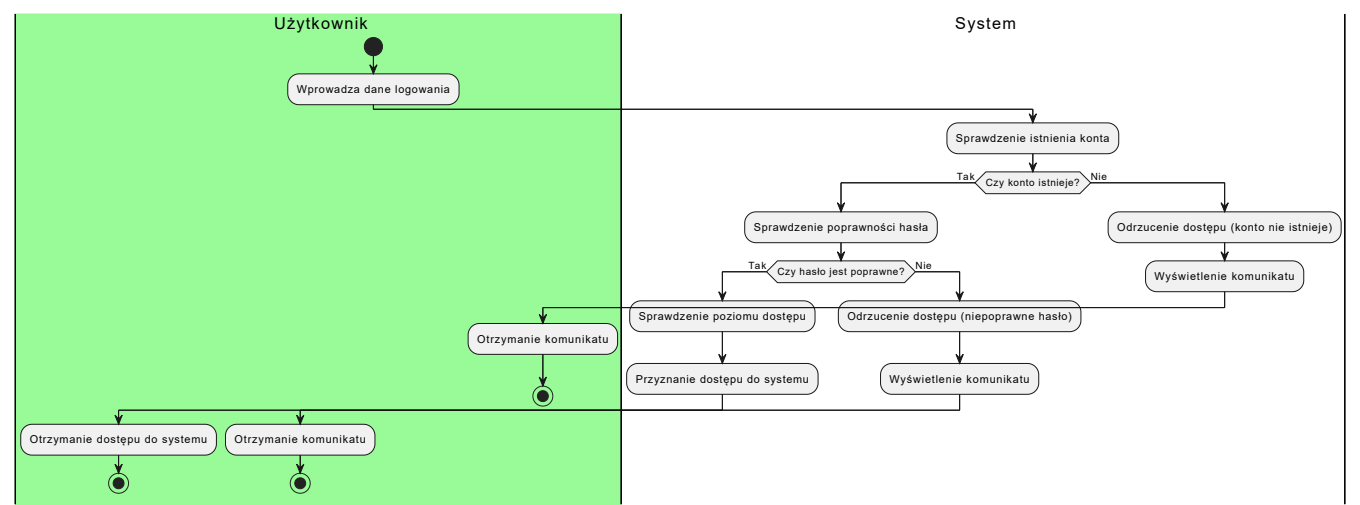
Kod diagramu


```
@startuml
|#palegreen|u| Użytkownik
|s| System

|u|
start
:Użytkownik wprowadza kryteria wyszukiwania;
|s|
:System filtruje dane na podstawie kryteriów;
|s|
:System prezentuje wyniki;
|u|
:Użytkownik otrzymuje wyniki;
stop
@enduml
```

(PB0005) Zarządzanie dostępem

Diagram



Opis

Na podstawie diagramu PlantUML, można opisać proces biznesowy związany z **zarządzaniem dostępem**. Proces ten składa się z kilku głównych etapów, w których uczestniczą dwie główne role:

1. Wprowadzenie danych logowania (User Inputs Credentials)

- Rola: Użytkownik (user)
- Opis: Użytkownik wprowadza swoje dane logowania .
- Kolor: #palegreen
- Oznaczenie: u

2. Sprawdzenie istnienia konta (Account Existence Check)

- Rola: System (system)
- Opis: System weryfikuje, czy konto użytkownika istnieje w bazie danych.
- Oznaczenie: **s**

3. Sprawdzenie poprawności hasła (Password Check)

- Rola: System (system)
- Opis: System sprawdza, czy wprowadzone hasło jest poprawne w stosunku do zapisanych danych.
- Oznaczenie: **s**

4. Sprawdzenie poziomu dostępu (Access Level Check)

- Rola: System (system)
- Opis: Jeśli dane logowania są poprawne, system sprawdza poziom dostępu użytkownika (np. admin, użytkownik standardowy).
- Oznaczenie: **s**

5. Przyznanie dostępu (Access Granted)

- Rola: System (system)
- Opis: System przyznaje dostęp do systemu użytkownikowi, jeśli dane logowania są poprawne i użytkownik ma odpowiednie uprawnienia.
- Oznaczenie: **s**

6. Odrzucenie dostępu (Access Denied)

- Rola: System (system)
- Opis: Jeśli konto nie istnieje, dane logowania są niepoprawne lub użytkownik nie ma odpowiednich uprawnień, dostęp zostaje odrzucony.
- Oznaczenie: **s**

7. Wyświetlenie komunikatu (Display Message)

- Rola: Użytkownik (user)
- Opis: Użytkownik otrzymuje komunikat o powodzeniu lub błędzie w zależności od wyniku logowania.
- Kolor: **#palegreen**
- Oznaczenie: **u**

Proces biznesowy

Proces biznesowy polega na weryfikacji danych użytkownika i przyznaniu dostępu do systemu. Ma na celu zapewnienie bezpieczeństwa i kontroli nad dostępem do zasobów systemu, zapobiegając nieautoryzowanemu dostępowi. Proces składa się z kilku kroków, w których użytkownik dostarcza dane, a system je weryfikuje, przyznając lub odrzucając dostęp.

Kod diagramu

```
@startuml
|#palegreen|u| Użytkownik
start
:Wprowadza dane logowania;

|s| System
:Sprawdzenie istnienia konta;
if (Czy konto istnieje?) then (Tak)
  :Sprawdzenie poprawności hasła;
  if (Czy hasło jest poprawne?) then (Tak)
    :Sprawdzenie poziomu dostępu;
    :Przyznanie dostępu do systemu;
|u| Użytkownik
  :Otrzymanie dostępu do systemu;
  stop
|s| System
  else (Nie)

    :Odrzucenie dostępu (niepoprawne hasło);

    :Wyświetlenie komunikatu;
|u| Użytkownik
    :Otrzymanie komunikatu;
    stop
|s| System
  endif
else (Nie)
  :Odrzucenie dostępu (konto nie istnieje);

  :Wyświetlenie komunikatu;
|u| Użytkownik
  :Otrzymanie komunikatu;
  stop
endif
@enduml
```

(PB0006) Prezentowanie zbiorów danych

Diagram



Opis

Na podstawie diagramu PlantUML, można opisać proces biznesowy związany z **prezentowaniem zbiorów danych**. Proces ten składa się z następujących etapów, w których uczestnicza jedna rola:

1. Zbieranie danych

- Rola: System (system)
- Opis: Dane są automatycznie pobierane z różnych źródeł przez system. Pracownik nie musi podejmować żadnych działań w tym zakresie.
- Oznaczenie: **S**

2. Oczyszczanie i standaryzacja danych

- Rola: System (system)
- Opis: System usuwa błędy, duplikaty oraz niekompletne rekordy. Dane są standaryzowane do jednolitego formatu zgodnego z wymaganiami analitycznymi.
- Oznaczenie: **S**

3. Analiza oraz agregacja danych:

- Rola: System (system)
- Opis: System analizuje dane w celu identyfikacji trendów, wzorców oraz anomalii. Możliwe jest grupowanie danych według określonych kategorii.
- Oznaczenie: **S**

4. Wyodrębnianie kluczowych wskaźników:

- Rola: System (system)

- Opis: Na podstawie analizy system generuje kluczowe wskaźniki, np. średnie wartości, odchylenia standardowe, rankingi.
- Oznaczenie: **S**

5. Tworzenie wizualizacji (raporty, wykresy, dashboardy):

- Rola: System (system)
- Opis: System generuje raporty, interaktywne wykresy i dashboardy w celu lepszej interpretacji danych. Dane mogą być prezentowane w tabelach, diagramach czy mapach ciepła.
- Oznaczenie: **S**

6. Dystrybucja zestawów danych

- Rola: System (system)
- Opis: System automatycznie udostępnia przetworzone dane użytkownikom poprzez aplikację webową, API lub eksport do plików (np. JSON, XML). Użytkownicy mogą pobierać raporty lub filtrować interesujące ich dane.
- Oznaczenie: **S**

Proces biznesowy polega na przekształceniu surowych danych w przejrzyste i wartościowe informacje, które mogą być używane do podejmowania decyzji, analizy trendów oraz monitorowania wskaźników wydajności.

Kod diagramu

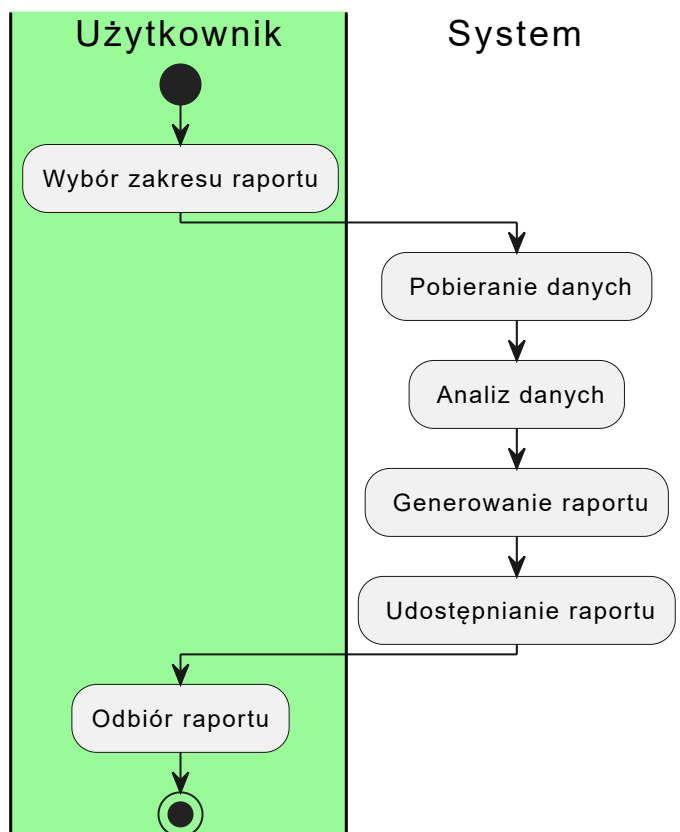
```
@startuml
|s| System

|s|
start
:Zbieranie danych;
:Oczyszczanie i standaryzacja danych;
:Analiza oraz agregacja danych;
:Wyodrębnianie kluczowych wskaźników;
:Tworzenie wizualizacji (raporty, wykresy, dashboardy);
:Dystrybucja zestawów danych;

stop
@enduml
```

(PB0007) Generowanie raportów o ruchu

Diagram



Opis

Na podstawie diagramu PlantUML, można opisać proces biznesowy związany z **generowaniem raportów o ruchu**:

1. Wybór zakresu raportu

- Rola: Użytkownik
- Opis: Użytkownik określa zakres danych, które mają być uwzględnione w raporcie, np. przedział czasowy, lokalizację, typy zdarzeń itp.
- Kolor: #palegreen
- Oznaczenie: u

2. Analiza danych

- Rola: System
- Opis: System przetwarza i analizuje dane dotyczące ruchu, agregując je, wykrywając anomalie oraz identyfikując trendy.

3. Generowanie raportu

- Rola: System
- Opis: System tworzy raport na podstawie zgromadzonych danych, który może zawierać wykresy, statystyki, mapy oraz inne wizualizacje ułatwiające interpretację wyników.

4. Udostępnienie raportu

- Rola: System

- Opis: Raport jest udostępniany użytkownikowi w wybranym formacie (np. PDF, CSV, JSON, interaktywna strona internetowa) i może być wysyłany na e-mail lub udostępniany przez API.

Proces biznesowy polega na automatycznym przetwarzaniu i analizie danych o ruchu, a następnie generowaniu raportów dostosowanych do potrzeb użytkownika. Dzięki temu organizacje mogą podejmować świadome decyzje oparte na rzeczywistych danych, optymalizować procesy i reagować na zmiany w ruchu w czasie rzeczywistym.

Kod diagramu

```
@startuml
|#palegreen|u|Użytkownik
|System|

|u|
start
:Wybór zakresu raportu;

|System|
: Pobieranie danych;

: Analiz danych;

: Generowanie raportu;

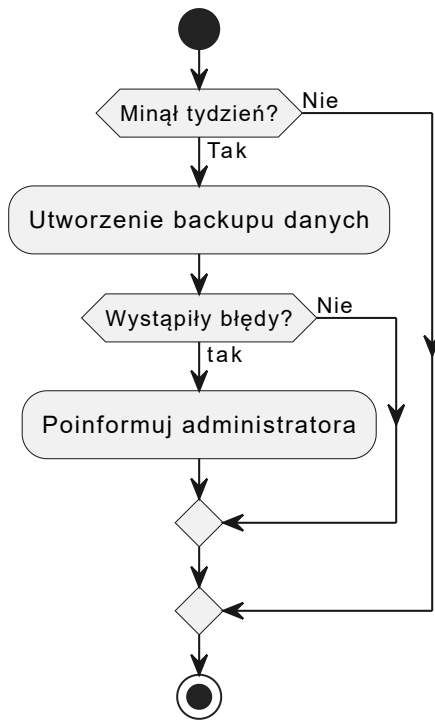
: Udostępnianie raportu;

|u|
:Odbiór raportu;

stop
@enduml
```

(PB0008) Regularna archiwizacja danych

Diagram



Opis

Na podstawie diagramu PlantUML, można opisać proces biznesowy związany z **regularną archiwizacją danych** :

1. Cykliczna aktywacja

- Rola: System
- System co tydzień aktywuje proces archiwizacji danych.

2. Utworzenie backupu danych

- Rola: System
- System tworzy kopię zapasową odzwierciedlającą aktualny stan bazy danych.

3. Poinformowanie administratora

- Rola: System
- System w przypadku wykrycia błędów podczas tworzenia kopii wysyła o tym wiadomość do administratora.

Proces biznesowy polega na regularnym tworzeniu kopii zapasowej aktualnego stanu danych w systemie. Celem tego procesu jest posiadanie możliwości cofnięcia się do poprzedniej wersji danych w przypadku poważniejszego błędu.

Kod diagramu

```
@startuml
start
```



```

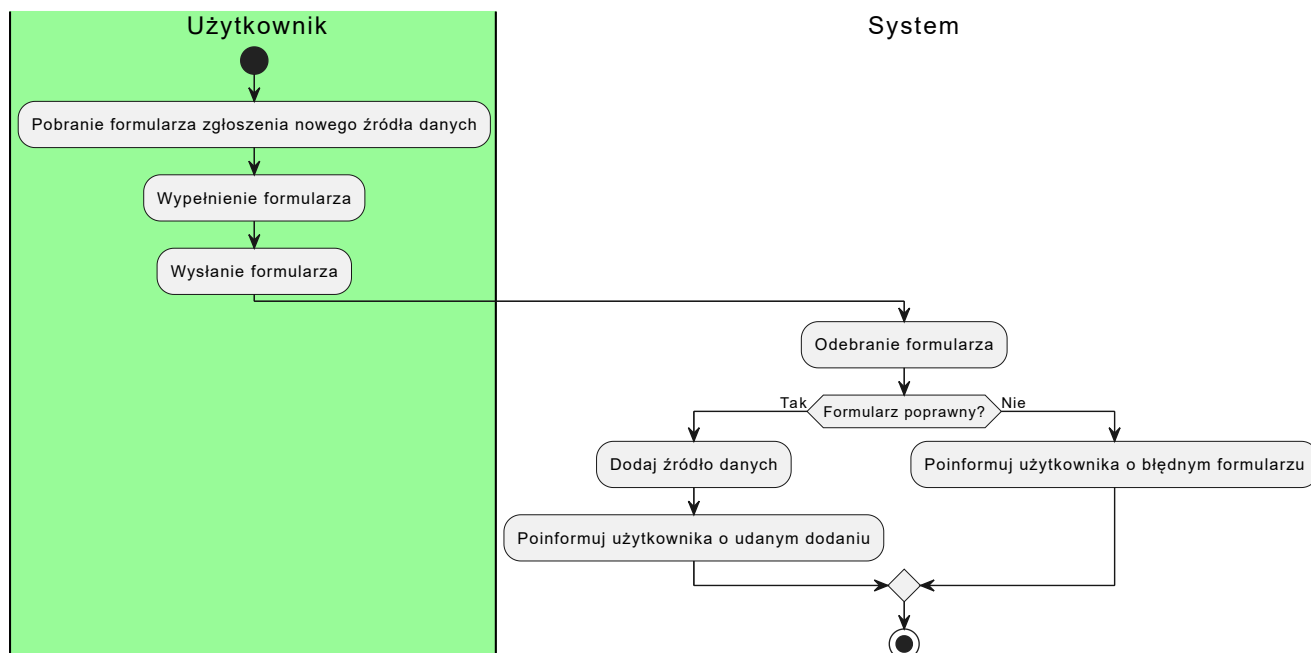
if (Minął tydzień?) then (Tak)
  :Utworzenie backupu danych;
  if (Wystąpiły błędy?) then (tak)
    :Poinformuj administratora;
  else (Nie)
  endif
endif
else (Nie)
endif

stop
@enduml

```

(PB0009) Zgłaszanie nowych źródeł danych

Diagram



Opis

Na podstawie diagramu PlantUML, można opisać proces biznesowy związany ze **zgłaszaniem nowych źródeł danych** :

1. Pobranie formularza

- Rola: Użytkownik
- Formularz zgłoszeniowy jest dostępny publicznie do pobrania.

2. Wypełnienie formularza

- Rola: Użytkownik
- Formularz należy uzupełnić odpowiednimi danymi które następnie będą walidowane

3. Walidacja formularza

- Rola: System
- System sprawdza poprawność formularza i w zależności od wyniku dodaje źródło danych lub informuje klienta o niepoprawnym formularzu.

Proces biznesowy polega na dodawaniu nowych źródeł danych za pomocą formularza. Celem tego procesu jest posiadanie możliwości szybkiego i sprawnego poszerzania naszej bazy danych.

Kod diagramu

```
@startuml
    |#palegreen|u| Użytkownik
    |s| System
    |u|
    start
    :Pobranie formularza zgłoszenia nowego źródła danych;
    :Wypełnienie formularza;
    :Wysłanie formularza;
    |s|
    :Odebranie formularza;
    if (Formularz poprawny?) then (Tak)
        :Dodaj źródło danych;
        :Poinformuj użytkownika o udanym dodaniu;
    else (Nie)
        :Poinformuj użytkownika o błędnym formularzu;
    endif
    stop
@enduml
```

(PB0010) Ocenianie zbiorów danych pod kątem przydatności

Diagram



Opis

Na podstawie diagramu PlantUML, można opisać proces biznesowy związany z **Ocenianiem zbiorów danych pod kątem przydatności**:

1. Wystawienie oceny

- Rola: Użytkownik
- Użytkownicy naszego serwisu mają możliwość oceniania źródeł danych pod kątem przydatności w 5 stopniowej skali (gwiazdki).

2. Weryfikacja unikalności

- Rola: System
- System weryfikuje czy użytkownik nie ocenił już danego zbioru danych.

3. Zaktualizowanie oceny

- Rola: System
- Wyświetlana ocena danego źródła jest średnią wszystkich poprzednich ocen oraz nowej dodanej.

Proces biznesowy polega na ocenieniu źródła danych. Celem tego procesu jest ułatwienie innym użytkownikom wyszukiwania wartościowych zbiorów danych.

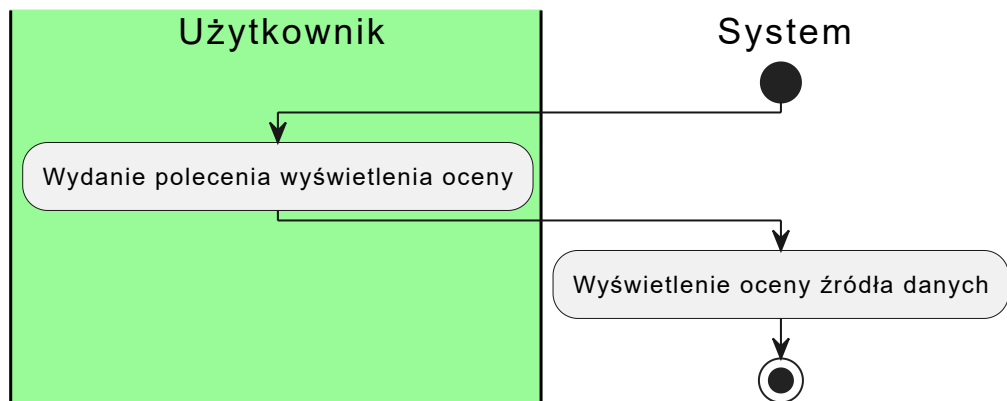
Kod diagramu

```
@startuml
    |#palegreen|u| Użytkownik
    |s| System
    |u|
    start
    :Wystawienie oceny dla danego źródła danych;
    |s|
    :Weryfikacja unikalności oceny;
    :Zaktualizowanie średniej ocen o nową ocenę;

    stop
@enduml
```

(PB0011) Prezentowanie ocen danych

Diagram



Opis

Na podstawie diagramu PlantUML, można opisać proces biznesowy związany z **prezentowaniem ocen danych** :

1. Wysłanie polecenia wyświetlenia oceny

- Rola: Użytkownik
- Użytkownik wyraża chęć zobaczenia oceny danego źródła danych.

2. Wyświetlenie oceny źródła danych

- Rola: System
- Każde źródło danych jest opisane swoją oceną, która jest średnią ocen wystawionych przez użytkowników w 5-stopniowej skali (gwiazdki).

Proces biznesowy polega na dodawaniu prezentowaniu oceny źródła danych. Celem tego procesu jest ułatwienie klientom rozpoznawania wartościowych zbiorów danych.

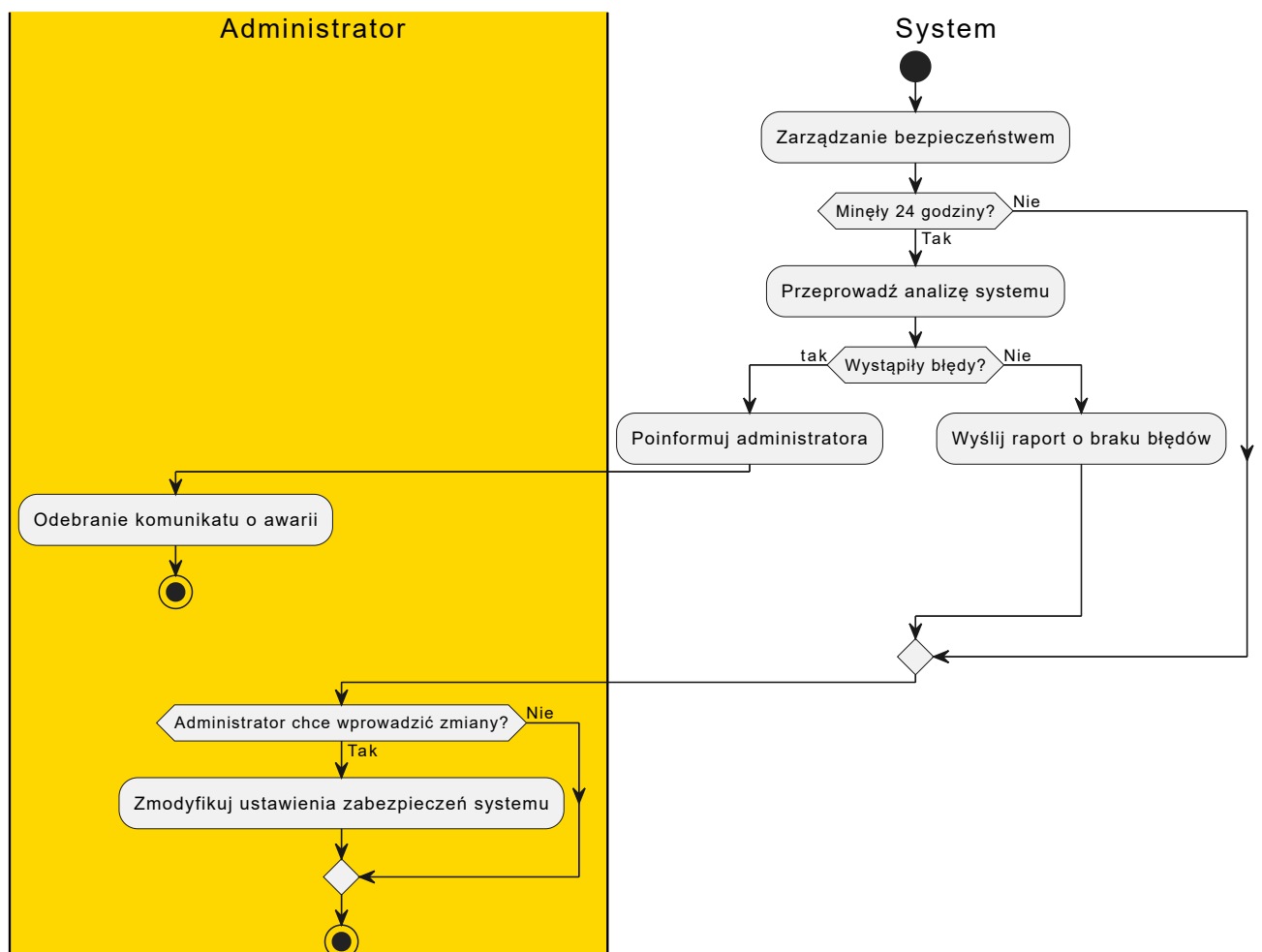
Kod diagramu

```
@startuml
    |#palegreen|u| Użytkownik
    |s| System

    start
    |u|
    :Wydanie polecenia wyświetlenia oceny;
    |s|
    :Wyświetlenie oceny źródła danych;

    stop
    @enduml
```

Diagram



Opis

Na podstawie diagramu PlantUML, można opisać proces biznesowy związany ze **zarządzaniem bezpieczeństwem** :

1. Generowanie raportu

- Rola: System
- System regularnie generuje raport o stanie.

2. Poinformowanie administratora

- Rola: System
- Administrator zostaje poinformowany o potencjalnym zagrożeniu i następnie podejmuje decyzje odnośnie kolejnych działań.

3. Modyfikowanie ustawień bezpieczeństwa

- Rola: Administrator
- Administrator może włączyć lub wyłączyć dane zabezpieczenia, jeżeli jest taka potrzeba.

Proces biznesowy polega na zarządzaniu bezpieczeństwem. Celem tego procesu jest odpowiednie reagowanie na sytuacje zagrożenia.

Kod diagramu

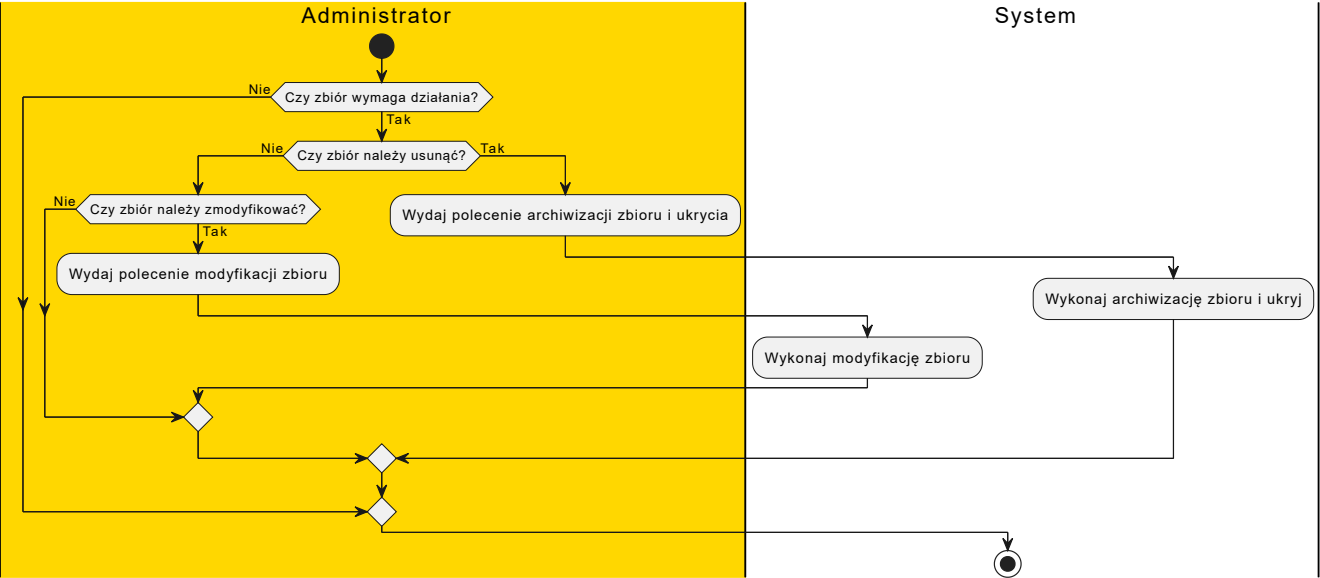
```
@startuml

|#gold|a| Administrator
|s| System
start
:Zarządzanie bezpieczeństwem;

if (Minęły 24 godziny?) then (Tak)
  :Przeprowadź analizę systemu;
  if (Wystąpiły błędy?) then (tak)
    :Poinformuj administratora;
    |a|
    :Odebranie komunikatu o awarii;
    stop
  else (Nie)
    |s|
    :Wyślij raport o braku błędów;
  endif
else (Nie)
endif
|a|
if (Administrator chce wprowadzić zmiany?) then (Tak)
  :Zmodyfikuj ustawienia zabezpieczeń systemu;
else (Nie)
endif

stop
@enduml
```

(PB0013) Zarządzanie zbiorami danych**Diagram**



Opis

Na podstawie diagramu PlantUML, można opisać proces biznesowy związany z **zarządzaniem zbiorami danych** :

1. Archiwizowanie zbioru

- Rola: System
- System dokonuje archiwizacji zbioru i ukrywa go.

2. Modyfikowanie zbioru

- Rola: System
- System realizuje odpowiednią modyfikację zbioru.

Proces biznesowy polega na zarządzaniu zbiorami danych. Celem tego procesu jest posiadanie spójnej i możliwie lekkiej bazy danych.

Kod diagramu

```
@startuml
|#gold|a| Administrator
|s| System
|a|

start
if (Czy zbiór wymaga działania?) then (Tak)
    if (Czy zbiór należy usunąć?) then (Nie)
        if (Czy zbiór należy zmodyfikować?) then (Nie)
            else (Tak)
                :Wydadaj polecenie modyfikacji zbioru;
                |s|
                :Wykonaj modyfikację zbioru;
            endif
        else (Tak)
            :Wydadaj polecenie archiwizacji zbioru i ukrycia;
        end
    end
end
if (Czy zbiór należy zmodyfikować?) then (Nie)
    :Wykonaj archiwizację zbioru i ukryj;
end
end
end
```

```

else (Tak)
|a|
:Wydadaj polecenie archiwizacji zbioru i ukrycia;
|s|
:Wykonaj archiwizację zbioru i ukryj;
endif
else (Nie)
endif

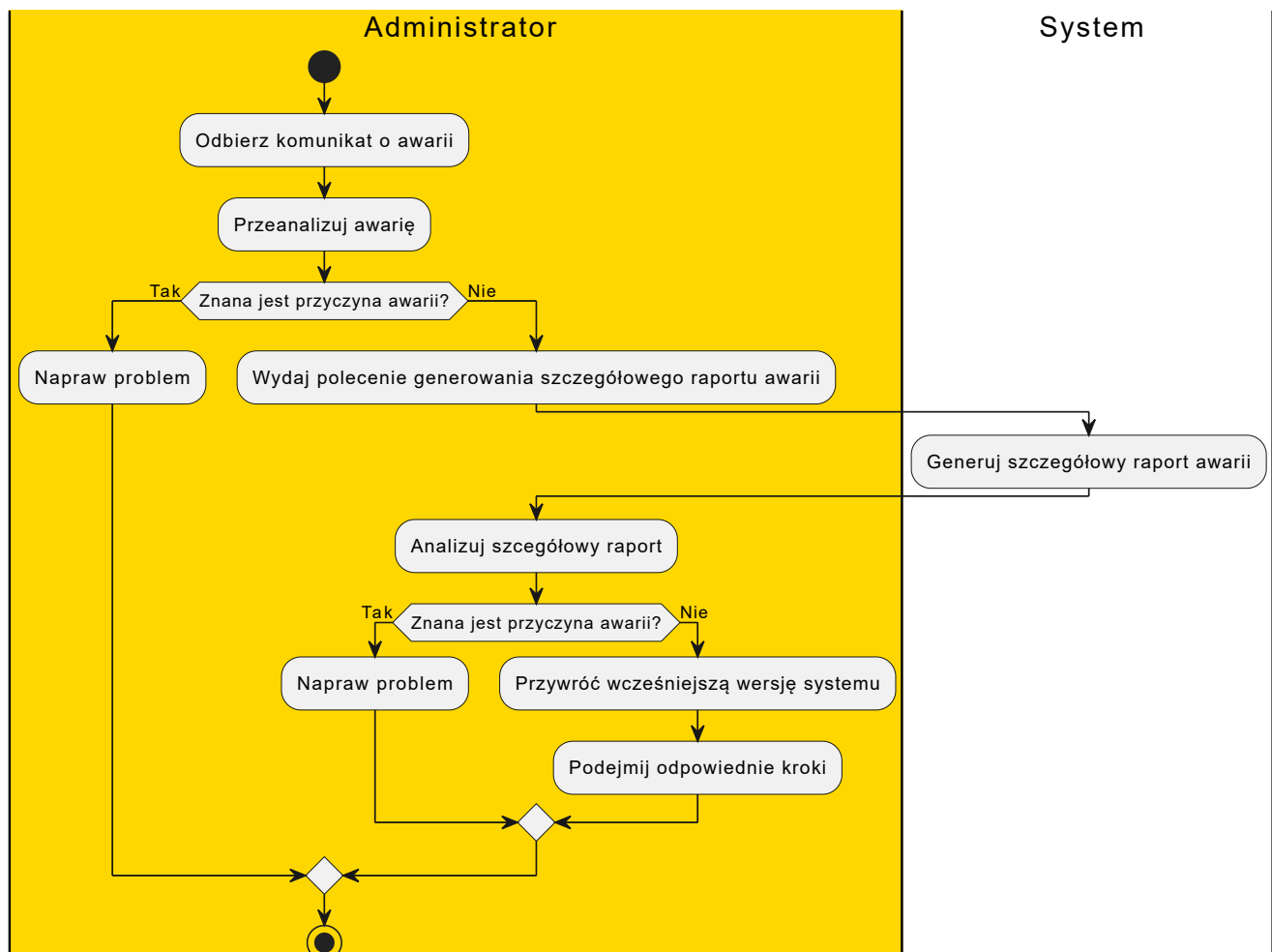
stop

@enduml

```

(PB0014) Informowanie o awariach

Diagram



Opis

Na podstawie diagramu PlantUML, można opisać proces biznesowy związany z **informowaniem o awariach** :

1. Odebranie komunikatu o awarii

- Rola: Administrator
- Administrator odbiera na swoim urządzeniu komunikat o awarii.

2. Przeanalizowanie awarii

- Rola: Administrator
- Administrator analizuje dostarczony mu ogólny raport o awarii.

3. Naprawienie problemu

- Rola: Administrator
- Jeżeli znana jest przyczyna awarii, administrator podejmuje odpowiednie działania w celu naprawienia błędu

4. Generowanie szczegółowego raportu

- Rola: System
- Jeżeli nie jest znana przyczyna awarii, system jest w stanie wygenerować szczegółowy raport dotyczący awarii

5. Przenalizowanie szczegółowego raportu

- Rola: Administrator
- Administrator analizuje szczegółowy raport dotyczący awarii

6. Przywrócenie wcześniejszej wersji systemu

- Rola: Administrator
- Administrator przywraca poprzednią, działającą wersję systemu.

7. Podjęcie odpowiednich kroków

- Rola: Administrator
- Jeżeli wciąż nie jest znana przyczyna, administrator może podjąć odpowiednie kroki jak np. poinformowanie o tym innych osób, wyłączenie niedziałającego modułu.

Proces biznesowy polega na informowaniu o awariach systemu. Celem tego procesu jest możliwie szybkie informowanie o awariach w celu efektywnej odpowiedzi na zaistniały problem.

Kod diagramu

```
@startuml
    |#gold|a| Administrator
    |s| System
    |a|
    start
    :Odbierz komunikat o awarii;
    :Przeanalizuj awarię;
    if (Znana jest przyczyna awarii?) then (Tak)
        :Napraw problem;
    else (Nie)
```

```
:Wyдай polecenie generowania szczegółowego raportu awarii;
|s|
:Generuj szczegółowy raport awarii;
|a|
:Analizuj szczegółowy raport;
if (Znana jest przyczyna awarii?) then (Tak)
  :Napraw problem;
else (Nie)
  :Przywróć wcześniejszą wersję systemu;
  :Podejmij odpowiednie kroki;
endif
endif

stop
@enduml
```

3. wizja systemu

Motto

"Otwarte dane, nieograniczone możliwości" "Łączymy świat z danymi publicznymi - prosto, bezpiecznie i efektywnie" "Jednolity dostęp do wiarygodnych danych publicznych dla wszystkich, zawsze i wszędzie"

3.1 Opisy problemów

Problemy

W tej sekcji przedstawiono kluczowe problemy, które system Zarządzania Otwartymi Danymi Publicznymi ma rozwiązać.

1. Rozproszenie zbiorów danych publicznych

Opis problemu: Obecnie dane publiczne są rozproszone w różnych systemach i platformach, co utrudnia ich odnalezienie i wykorzystanie.

Wpływ na interesariuszy:

- Klienci/Użytkownicy danych muszą przeszukiwać wiele źródeł, co jest czasochłonne i nieefektywne
- Programiści API nie mogą łatwo integrować swoich aplikacji z wieloma różnymi systemami
- Researcherzy/Analitycy danych mają problem z kompleksowym dostępem do potrzebnych zbiorów

Potrzeby wynikające z problemu:

- Centralne repozytorium danych publicznych
- Ujednolicony interfejs dostępu
- Spójny system metadanych opisujących zbiory

2. Niska jakość i nieaktualność danych

Opis problemu: Dostępne zbiory danych często zawierają błędy, niespójności lub są nieaktualne, co obniża ich wartość i przydatność.

Wpływ na interesariuszy:

- Klienci/Użytkownicy danych nie mogą polegać na otrzymanych informacjach
- Decydenci otrzymują niepełne lub błędne informacje utrudniające podejmowanie decyzji
- Dostawcy danych tracą wiarygodność

Potrzeby wynikające z problemu:

- Mechanizmy weryfikacji poprawności danych
- System regularnych aktualizacji
- Narzędzia do wykrywania anomalii i niespójności
- Możliwość zgłaszania błędów przez użytkowników

3. Trudności w zarządzaniu zbiorami danych

Opis problemu: Zarządzanie dużymi zbiorami danych jest skomplikowane, szczególnie gdy pochodzą z różnych źródeł i mają różne formaty.

Wpływ na interesariuszy:

- Administratorzy systemu zmagają się z chaotycznym systemem organizacji danych
- Dostawcy danych muszą ręcznie konwertować dane do odpowiednich formatów
- Archiwiści danych mają problem z utrzymaniem spójności i wersjonowaniem zbiorów

Potrzeby wynikające z problemu:

- Efektywne narzędzia do zarządzania metadanymi
- Automatyzacja procesów importu i eksportu danych
- System zarządzania schematami danych

4. Problemy z bezpieczeństwem i kontrolą dostępu

Opis problemu: Brak odpowiednich mechanizmów zabezpieczeń oraz kontroli dostępu do wrażliwych danych.

Wpływ na interesariuszy:

- Audytorzy/Inspektorzy Ochrony Danych zgłaszają niezgodność z przepisami
- Kierownictwo instytucji obawia się naruszeń bezpieczeństwa i konsekwencji prawnych
- Użytkownicy nie mają pewności, czy ich dane osobowe są odpowiednio chronione

Potrzeby wynikające z problemu:

- Mechanizmy uwierzytelniania i autoryzacji
- Szyfrowanie wrażliwych danych
- System śledzenia dostępu i zmian

5. Nieefektywne wyszukiwanie i filtrowanie danych

Opis problemu: Istniejące systemy nie zapewniają zaawansowanych możliwości wyszukiwania i filtrowania danych.

Wpływ na interesariuszy:

- Klienci/Użytkownicy danych tracą czas na ręczne przeszukiwanie zbiorów
- Researcherzy/Analitycy danych nie mogą szybko dotrzeć do potrzebnych informacji
- Programiści API mają ograniczone możliwości przeszukiwania danych przez API

Potrzeby wynikające z problemu:

- Zaawansowane mechanizmy wyszukiwania i filtrowania
- Indeksowanie danych zwiększające wydajność wyszukiwania
- Intuicyjny interfejs wyszukiwania

6. Brak integracji z istniejącymi systemami

Opis problemu: Trudności w integracji nowego systemu z istniejącą infrastrukturą i systemami zewnętrznymi.

Wpływ na interesariuszy:

- Dział IT musi zarządzać wieloma niepowiązanymi systemami
- Administratorzy systemu wykonują podwójną pracę, wprowadzając te same dane do różnych systemów
- Dostawcy danych muszą dublować operacje w różnych systemach

Potrzeby wynikające z problemu:

- Integracja z platformą CKAN
- Dobrze udokumentowane API
- Mechanizmy synchronizacji danych

7. Ograniczone możliwości prezentacji i wizualizacji danych

Opis problemu: Dane są często prezentowane w surowej formie, bez odpowiednich narzędzi do ich wizualizacji i analizy.

Wpływ na interesariuszy:

- Klienci/Użytkownicy danych mają trudności ze zrozumieniem złożonych zbiorów
- Decydenci nie mogą szybko uzyskać potrzebnych informacji w przystępnej formie
- Researcherzy/Analitycy danych muszą używać zewnętrznych narzędzi do wizualizacji

Potrzeby wynikające z problemu:

- Narzędzia do wizualizacji danych
- Generowanie raportów w różnych formatach
- Prezentacja danych w formie wykresów i tabel

8. Wysokie koszty utrzymania systemów danych

Opis problemu: Utrzymanie wielu rozproszonych systemów generuje wysokie koszty operacyjne i wymaga dużych zasobów IT.

Wpływ na interesariuszy:

- Kierownictwo instytucji zmaga się z rosnącymi kosztami IT

- Dział IT musi dzielić zasoby między wiele systemów
- Zespół deweloperski spędza czas na utrzymaniu zamiast na rozwoju

Potrzeby wynikające z problemu:

- Optymalizacja wykorzystania zasobów
- Automatyzacja rutynowych operacji
- Skalowalność systemu

9. Brak mechanizmów do zbierania informacji zwrotnej

Opis problemu: Istniejące systemy nie umożliwiają użytkownikom oceniania i komentowania zbiorów danych.

Wpływ na interesariuszy:

- Dostawcy danych nie otrzymują informacji o użyteczności udostępnianych zbiorów
- Klienci/Użytkownicy danych nie mogą dzielić się spostrzeżeniami i wskazywać problemów
- Kierownictwo instytucji nie ma informacji zwrotnej o wartości udostępnianych danych

Potrzeby wynikające z problemu:

- System ocen i komentarzy
- Mechanizmy zbierania i analizy opinii użytkowników
- Możliwość zgłaszania błędów w danych

10. Niedostosowanie do potrzeb osób z niepełnosprawnościami

Opis problemu: Obecne systemy często nie spełniają standardów dostępności dla osób z niepełnosprawnościami.

Wpływ na interesariuszy:

- Klienci/Użytkownicy z niepełnosprawnościami są wykluczeni z dostępu do danych publicznych
- Audytorzy/Inspektorzy zgłaszają niezgodność z regulacjami dotyczącymi dostępności
- Kierownictwo instytucji narażone jest na konsekwencje prawne

Potrzeby wynikające z problemu:

- Interfejs zgodny ze standardami WCAG
- Kompatybilność z technologiami wspomagającymi
- Alternatywne metody dostępu do danych

3.2 Interesariusze

Interesariusze systemu

W tej sekcji przedstawiono osoby i grupy, które są bezpośrednio lub pośrednio zainteresowane powstaniem systemu Zarządzania Otwartymi Danymi Publicznymi i na które wdrożenie tego systemu będzie miało wpływ.

1. Administrator systemu

Typ interesariusza: Użytkownik wewnętrzny, personel techniczny

Opis: Osoby odpowiedzialne za zarządzanie systemem, rejestrowanie nowych źródeł danych, otrzymywanie powiadomień o błędach oraz tworzenie kopii zapasowych.

Zakres obowiązków:

- Zarządzanie konfiguracją systemu i zbiorami danych
- Monitorowanie wydajności i dostępności
- Zarządzanie kontami użytkowników i uprawnieniami
- Rozwiązywanie problemów technicznych

Kryteria zadowolenia:

- Intuicyjny panel administracyjny
- Narzędzia do automatycznego monitorowania
- Efektywne zarządzanie użytkownikami

Zadania w projekcie:

- Konsultowanie wymagań technicznych
- Testowanie funkcji administracyjnych
- Przygotowanie procedur backupu

2. Dostawca danych

Typ interesariusza: Użytkownik wewnętrzny, ekspert dziedzinowy

Opis: Pracownicy instytucji dostarczający dane do systemu, posiadający wiedzę ekspercką o tych danych.

Zakres obowiązków:

- Przygotowywanie danych do publikacji
- Weryfikacja poprawności danych
- Aktualizowanie zbiorów danych
- Zarządzanie metadanymi i schematami

Kryteria zadowolenia:

- Łatwy proces dodawania danych
- Automatyczna walidacja zgodności
- Wsparcie dla różnych formatów

Zadania w projekcie:

- Definiowanie wymagań dotyczących formatów
- Testowanie funkcji zarządzania danymi
- Dostarczanie przykładowych zbiorów

3. Klient/Użytkownik danych

Typ interesariusza: Użytkownik końcowy

Opis: Osoby korzystające z systemu w celu uzyskania informacji publicznych (obywatele, dziennikarze, aktywiści, przedsiębiorcy).

Zakres obowiązków:

- Wyszukiwanie i przeglądanie danych
- Ocenianie wartości zbiorów danych
- Zgłaszanie problemów z danymi

Kryteria zadowolenia:

- Intuicyjny interfejs użytkownika
- Łatwość wyszukiwania i filtrowania
- Czytelna prezentacja w różnych formatach
- Dostępność na różnych urządzeniach

Zadania w projekcie:

- Udział w badaniach użyteczności
- Testowanie interfejsu
- Zgłaszanie sugestii

4. Programista API

Typ interesariusza: Klient zewnętrzny, użytkownik techniczny

Opis: Deweloperzy tworzący aplikacje wykorzystujące dane z systemu poprzez API.

Zakres obowiązków:

- Integracja swoich aplikacji z API systemu
- Tworzenie usług bazujących na danych publicznych

Kryteria zadowolenia:

- Dobrze udokumentowane API
- Stabilność i wydajność interfejsu
- Narzędzia do testowania

Zadania w projekcie:

- Konsultowanie specyfikacji API
- Testowanie prototypów

5. Researcher / Analityk danych

Typ interesariusza: Użytkownik zewnętrzny

Opis: Naukowcy, badacze i analitycy wykorzystujący dane do celów badawczych i edukacyjnych.

Zakres obowiązków:

- Analiza i interpretacja danych publicznych

- Tworzenie modeli i wizualizacji
- Publikowanie wyników badań

Kryteria zadowolenia:

- Dostęp do kompletnych zbiorów
- Zaawansowane opcje wyszukiwania
- Możliwość pobierania w formatach analitycznych

Zadania w projekcie:

- Definiowanie wymagań analitycznych
- Ocena przydatności danych do badań

6. Kierownictwo instytucji / Decydenci

Typ interesariusza: Zamawiający, osoba decyzyjna

Opis: Kadra kierownicza odpowiedzialna za strategiczne decyzje i odbierająca dane wyżej w hierarchii.

Zakres obowiązków:

- Decyzje strategiczne o zakresie danych
- Zatwierdzanie budżetu
- Nadzór nad zgodnością z regulacjami

Kryteria zadowolenia:

- Zgodność z przepisami
- Pozytywny wpływ wizerunkowy
- Mierzalne korzyści z udostępniania danych

Zadania w projekcie:

- Zatwierdzanie specyfikacji
- Zapewnienie zasobów
- Ustalanie priorytetów

7. Zespół deweloperski

Typ interesariusza: Personel zaangażowany w tworzenie systemu

Opis: Programiści i projektanci pracujący nad budową systemu.

Zakres obowiązków:

- Implementacja funkcjonalności
- Testowanie i debugowanie
- Integracja z zewnętrznymi systemami

Kryteria zadowolenia:

- Jasne wymagania

- Dostęp do zasobów
- Możliwość wykorzystania nowych technologii

Zadania w projekcie:

- Implementacja funkcji
- Projektowanie interfejsu
- Dokumentacja techniczna

8. Dział IT instytucji

Typ interesariusza: Użytkownik wewnętrzny, personel techniczny

Opis: Zespół odpowiedzialny za integrację systemu z istniejącą infrastrukturą.

Zakres obowiązków:

- Zapewnienie infrastruktury
- Integracja z systemami instytucji
- Wsparcie techniczne pierwszej linii

Kryteria zadowolenia:

- Kompatybilność z infrastrukturą
- Łatwość utrzymania
- Solidna dokumentacja

Zadania w projekcie:

- Konsultowanie wymagań infrastrukturalnych
- Przygotowanie środowiska

9. Audytor / Inspektor Ochrony Danych

Typ interesariusza: Specjalista ds. zgodności

Opis: Osoby weryfikujące zgodność systemu z przepisami prawa i standardami bezpieczeństwa.

Zakres obowiązków:

- Ocena zgodności z RODO, NIS2
- Monitorowanie procesów przetwarzania
- Przeprowadzanie audytów bezpieczeństwa

Kryteria zadowolenia:

- Przejrzysta dokumentacja procesów
- Mechanizmy zapewniające zgodność
- Ślad audytowy operacji

Zadania w projekcie:

- Weryfikacja zgodności rozwiązań

- Ocena wpływu na ochronę danych

10. Archiwista danych

Typ interesariusza: Użytkownik wewnętrzny

Opis: Osoby odpowiedzialne za długoterminowe przechowywanie i wersjonowanie danych.

Zakres obowiązków:

- Zarządzanie archiwizacją danych
- Utrzymywanie historii danych
- Zarządzanie politykami retencji

Kryteria zadowolenia:

- Skuteczne mechanizmy archiwizacji
- Możliwość odtworzenia historycznych wersji
- Efektywne zarządzanie przestrzenią

Zadania w projekcie:

- Definiowanie strategii archiwizacji
- Testowanie mechanizmów wersjonowania

11. Reklamodawcy

Typ interesariusza: Partner biznesowy

Opis: Firmy zainteresowane dotarciem do odbiorców systemu poprzez reklamy.

Zakres obowiązków:

- Dostarczanie materiałów reklamowych
- Opłacanie usług reklamowych

Kryteria zadowolenia:

- Przejrzyste raportowanie skuteczności
- Dotarcie do właściwej grupy docelowej
- Pozytywny zwrot z inwestycji

Zadania w projekcie:

- Konsultacja modelu monetyzacji
- Testowanie platformy reklamowej

3.3 Cechy funkcjonalne

Cechy funkcjonalne systemu

System Zarządzania Otwartymi Danymi Publicznymi powinien realizować następujące funkcjonalności, pogrupowane według stopnia ważności.

Priorytet wysoki

1. Zarządzanie metadanymi udostępnionych zbiorów danych

- System musi umożliwiać przechowywanie i zarządzanie metadanymi opisującymi zbiory danych.
- Metadane powinny zawierać informacje o źródle danych, dacie utworzenia, aktualizacji, formacie, itp.

2. Zarządzanie dystrybucjami zbiorów danych i ich powiązanie ze źródłami danych

- System powinien umożliwiać zarządzanie różnymi dystrybucjami tych samych zbiorów danych.
- Musi istnieć możliwość śledzenia pochodzenia danych i powiązania ich ze źródłami.

3. Możliwość wygodnego wprowadzania, przeglądania, filtrowania i wyszukiwania zbiorów danych i ich dystrybucji

- System powinien zapewniać intuicyjny interfejs do wprowadzania nowych danych.
- Użytkownicy muszą mieć możliwość efektywnego przeszukiwania i filtrowania zbiorów danych.

4. Zarządzanie schematami danych i przypisywanie schematów danych do zbiorów danych

- System musi umożliwiać definiowanie schematów danych.
- Użytkownik powinien móc przypisywać schematy do odpowiednich zbiorów danych.

5. Weryfikacja zgodności dystrybucji zbiorów danych ze schematami

- System powinien automatycznie weryfikować, czy dane są zgodne z przypisanymi schematami.
- W przypadku wykrycia niezgodności, system musi informować o problemach.

6. Import i eksport danych w różnych formatach z zachowaniem zgodności ze schematami

- System musi obsługiwać import danych z różnych źródeł i w różnych formatach.
- Eksport danych powinien być możliwy w wielu formatach, z zachowaniem zgodności ze schematami.

7. Integracja z systemem CKAN

- System powinien integrować się z platformą CKAN do zarządzania danymi.
- Integracja powinna umożliwiać wymianę danych między systemami.

8. Filtrowanie i wyszukiwanie danych

- Użytkownicy powinni móc filtrować dane według różnych kryteriów.
- System musi zapewniać zaawansowane mechanizmy wyszukiwania.

9. Autentykacja danych

- System musi weryfikować autentyczność danych.
- Powinien istnieć mechanizm weryfikacji źródła danych.

10. Autoryzacja użytkowników

- System musi posiadać mechanizm zarządzania uprawnieniami użytkowników.

- Różni użytkownicy powinni mieć różne poziomy dostępu.

11. Przyznawanie dostępu

- Administrator powinien móc przyznawać i odbierać dostęp do systemu.
- System musi umożliwiać definiowanie ról i uprawnień.

12. Sprawdzenie poziomu dostępu użytkownika

- System musi weryfikować uprawnienia użytkownika przed wykonaniem operacji.
- Poziomy dostępu powinny być sprawdzane w czasie rzeczywistym.

13. Dodawanie danych przez API

- System musi udostępniać API do automatycznego dodawania danych.
- API powinno być dobrze udokumentowane i zabezpieczone.

14. Prezentowanie wszystkich zbiorów danych

- System musi prezentować listę wszystkich dostępnych zbiorów danych.
- Prezentacja powinna być przejrzysta i umożliwiać sortowanie.

15. Ulepszanie jakości danych

- System powinien posiadać mechanizmy do poprawy jakości przechowywanych danych.
- Należy implementować procesy weryfikacji i czyszczenia danych.

16. Utrzymanie aktualności danych

- System musi zapewniać mechanizmy regularnego aktualizowania danych.
- Użytkownicy powinni być informowani o statusie aktualności danych.

17. Wykrywanie anomalii

- System powinien automatycznie wykrywać anomalie i niespójności w danych.
- O wykrytych anomalii system powinien informować odpowiedzialnych użytkowników.

18. Zapewnienie bezpieczeństwa danych

- System musi implementować mechanizmy zabezpieczające dane przed nieautoryzowanym dostępem.
- Należy stosować odpowiednie mechanizmy kryptograficzne i kontroli dostępu.

19. Wykrywanie błędów w zbiorach

- System powinien automatycznie wykrywać błędy w zbiorach danych.
- Wykryte błędy powinny być raportowane i oznaczane do korekty.

20. Szyfrowanie danych wrażliwych

- Dane wrażliwe muszą być przechowywane w formie zaszyfrowanej.
- System powinien automatycznie rozpoznawać i szyfrować wrażliwe informacje.

21. Szyfrowanie przesyłu

- Komunikacja z systemem musi być szyfrowana.
- Należy stosować aktualne protokoły kryptograficzne do zabezpieczenia transmisji.

22. Automatyczna cykliczna walidacja zbiorów

- System powinien cyklicznie przeprowadzać walidację zbiorów danych.
- Wyniki walidacji powinny być raportowane odpowiednim użytkownikom.

Priorytet średni

1. System ocen oraz uwag od użytkowników

- Użytkownicy powinni móc oceniać zbiory danych i pozostawiać komentarze.
- System powinien agregować i wyświetlać oceny.

2. Prezentacja wyników wyszukiwania w różnych formatach

- System powinien prezentować wyniki wyszukiwania w różnych formatach (tabele, wykresy).
- Użytkownik powinien móc wybierać sposób prezentacji.

3. Wyodrębnianie kluczowych wskaźników

- System powinien automatycznie analizować dane i wyodrębniać kluczowe wskaźniki.
- Wskaźniki powinny być prezentowane w czytelny sposób.

4. Zgłaszanie błędów danych

- Użytkownicy powinni móc zgłaszać błędy w danych.
- System powinien zarządzać zgłoszeniami i umożliwiać ich weryfikację.

5. Generowanie raportów

- System musi umożliwiać generowanie raportów na podstawie danych.
- Raporty powinny być dostępne w różnych formatach.

6. Przedstawienie historii zmian w przystępny sposób

- System powinien śledzić historię zmian w danych.
- Historia zmian powinna być prezentowana w sposób czytelny dla użytkownika.

7. Powiadomienia o aktualizacjach

- System powinien informować użytkowników o aktualizacjach zbiorów danych.
- Użytkownicy powinni móc subskrybować powiadomienia dla określonych zbiorów.

8. Integracja narzędzi do wizualizacji danych

- System powinien integrować narzędzia umożliwiające wizualizację danych.
- Wizualizacje powinny być interaktywne i konfigurowalne.

9. Aktualizowanie wybranych zbiorów co sekundę

- System powinien umożliwiać częste aktualizacje wybranych zbiorów danych.
- Mechanizm aktualizacji powinien być wydajny i nie obciążać systemu.

10. Udostępnianie danych wyżej w hierarchii

- System powinien umożliwiać udostępnianie wybranych zbiorów danych na wyższe poziomy hierarchii.
- Proces udostępniania powinien być kontrolowany i zabezpieczony.

11. Archiwizacja danych

- System powinien automatycznie archiwizować starsze dane.
- Dane archiwalne powinny być dostępne na żądanie.

12. Wersjonowanie danych

- System musi umożliwiać przechowywanie różnych wersji tych samych zbiorów danych.
- Użytkownicy powinni mieć możliwość porównania różnych wersji danych.

13. Przechowywanie logów systemu w celu audytu

- System powinien przechowywać logi wszystkich operacji.
- Logi powinny być dostępne na potrzeby audytu i analizy bezpieczeństwa.

14. Zbieranie i analizowanie ocen użytkowników

- System powinien zbierać opinie użytkowników o danych i funkcjonalnościach.
- Zebrane opinie powinny być analizowane w celu poprawy jakości systemu.

15. Automatyzacja aktualizacji

- System powinien umożliwiać automatyczne aktualizacje danych z zewnętrznych źródeł.
- Proces aktualizacji powinien być monitorowany i raportowany.

16. Analiza wykorzystania danych

- System powinien analizować, które zbiory danych są najczęściej używane.
- Analiza powinna dostarczać informacji o wzorcach korzystania z danych.

17. Wykrywanie zagrożeń bezpieczeństwa

- System powinien monitorować próby nieautoryzowanego dostępu.
- O potencjalnych zagrożeniach powinni być informowani administratorzy.

18. Monitorowanie wydajności

- System powinien monitorować swoją wydajność i obciążenie.
- W przypadku problemów wydajnościowych powinny być generowane alerty.

Priorytet niski

1. Zapewnienie tłumaczenia na inne języki

- System powinien wspierać wielojęzyczność.
- Interfejs i dane powinny być dostępne w różnych językach.

2. Statystyki wyszukiwań

- System powinien zbierać statystyki dotyczące wyszukiwań.
- Statystyki powinny być dostępne dla administratorów.

3. Ocena UI

- Użytkownicy powinni móc oceniać interfejs użytkownika.
- System powinien zbierać opinie na temat użyteczności.

4. Śledzenie aktywności użytkownika w ocenianiu

- System powinien śledzić aktywność użytkowników związaną z ocenianiem zbiorów.
- Informacje te powinny być wykorzystywane do poprawy jakości danych.

5. Ocenianie danych pod kątem przydatności

- Użytkownicy powinni móc oceniać przydatność danych.
- System powinien uwzględniać te oceny w algorytmach wyszukiwania.

6. Zarabianie na reklamach

- System może posiadać mechanizmy monetyzacji poprzez wyświetlanie reklam.
- Reklamy powinny być nieinwazyjne i związane z kontekstem.

7. Wspieranie badań naukowych poprzez dostarczanie zbiorów

- System powinien oferować specjalne funkcje dla użytkowników akademickich.
- Dane powinny być przygotowane w formacie odpowiednim dla celów badawczych.

8. Cachowanie odpowiedzi

- System powinien implementować mechanizmy cachowania dla częstych zapytań.
- Cachowanie powinno optymalizować wydajność i zmniejszać obciążenie bazy danych.

3.4 Cechy jakościowe

Cechy jakościowe systemu

System Zarządzania Otwartymi Danymi Publicznymi powinien spełniać następujące wymagania jakościowe, pogrupowane według priorytetu.

Priorytet wysoki

1. Efektywne zarządzanie zbiorami danych

- System powinien umożliwiać efektywne i ergonomiczne zarządzanie zbiorami danych.
- Interfejs zarządzania powinien być intuicyjny i minimalizować liczbę kroków potrzebnych do wykonania operacji.

2. Zapewnienie stałego dostępu do danych

- System musi zapewniać nieprzerwany dostęp do danych z minimalną liczbą niedostępności.
- Architektura powinna być redundantna, aby zapobiegać przestojom.

3. Time to recovery

- W przypadku awarii, czas przywracania funkcjonalności systemu nie powinien przekraczać 2 godzin.
- Proces odzyskiwania powinien być zautomatyzowany i regularnie testowany.

4. Ochrona przed atakami

- System powinien posiadać zabezpieczenia przed typowymi atakami sieciowymi (DDoS, SQL Injection, XSS).
- Regularne testy penetracyjne i audyty bezpieczeństwa powinny być przeprowadzane.

5. Ciągłość działania

- System powinien działać bez przerw, z minimalnymi planowanymi przerwami technicznymi.
- Aktualizacje systemu powinny odbywać się bez całkowitego wyłączenia usługi.

6. Przejrzystość interfejsu

- Interfejs użytkownika powinien być intuicyjny i łatwy w obsłudze dla użytkowników o różnym poziomie zaawansowania.
- Elementy interfejsu powinny być logicznie pogrupowane i opisane.

7. Sprawne dodawanie nowych źródeł danych

- Proces dodawania nowych źródeł danych powinien być prosty i szybki w implementacji.
- Dodawanie nowego źródła powinno wymagać minimalnej konfiguracji.

8. Zmniejszenie kosztów działania

- System powinien optymalizować wykorzystanie zasobów, aby minimalizować koszty operacyjne.
- Automatyzacja rutynowych operacji powinna redukować koszty utrzymania.

Priorytet średni

1. SLA (poziom dostępności)

- System powinien zapewniać poziom dostępności (SLA) na poziomie 99.99999% (tzw. "seven nines").
- Monitoring dostępności powinien umożliwiać szybką reakcję na problemy.

2. Zminimalizowanie czasu wyszukiwania

- Czas potrzebny na wyszukanie danych powinien być jak najkrótszy.
- Maksymalny czas przeszukiwania zbiorów danych nie powinien przekraczać 20 sekund, nawet przy złożonych zapytaniach.

3. Skalowalność systemu

- Architektura powinna umożliwiać skalowanie poziome i pionowe w zależności od rosnącego obciążenia.
- System powinien automatycznie dostosowywać wykorzystanie zasobów do bieżącego zapotrzebowania.

4. Minimalizacja opóźnień

- System powinien być zoptymalizowany pod kątem minimalizacji opóźnień w dostępie do danych.
- Opóźnienia w przesyłaniu i przetwarzaniu danych powinny być monitorowane i optymalizowane.

5. Wsparcie dla różnych formatów

- System powinien obsługiwać szeroki zakres formatów danych wejściowych i wyjściowych.
- Konwersja między formatami powinna odbywać się bez utraty informacji.

6. Adaptacyjność UI

- Interfejs powinien być w pełni responsywny i dostosowywać się do różnych rozmiarów ekranów (desktop, tablet, telefon).
- Układ elementów interfejsu powinien automatycznie reagować na zmianę rozdzielczości ekranu.

7. Efektywność wprowadzania danych

- Proces wprowadzania nowych danych nie powinien zajmować więcej niż 2 minuty.
- Interfejs wprowadzania danych powinien oferować podpowiedzi i automatyczne uzupełnianie.

8. Archiwizacja danych

- Dane historyczne powinny być archiwizowane w sposób efektywny, nie obciążający głównej bazy danych.
- System archiwizacji powinien umożliwiać szybkie przywracanie archiwalnych danych w razie potrzeby.

9. Szybkość powiadomień

- Czas od wysłania do otrzymania powiadomienia nie powinien przekraczać 5 minut.
- System powinien optymalizować dostarczanie powiadomień w zależności od ich priorytetu.

10. Szybkość aktualizacji danych

- Wybrane, krytyczne zbiory danych powinny być aktualizowane w czasie rzeczywistym, z opóźnieniem nieprzekraczającym jednej sekundy.
- Mechanizm aktualizacji powinien być wydajny i nie obciążać nadmiernie systemu.

Priorytet niski

1. Śledzenie aktywności użytkowników

- System powinien śledzić aktywność użytkowników związaną z ocenianiem zbiorów danych.
- Dane o aktywności powinny być analizowane w celu poprawy jakości systemu.

2. Szybkość generowania raportów

- Generowanie standardowych raportów powinno zajmować mniej niż jedną minutę.
- System powinien optymalizować proces generowania raportów poprzez wstępne przetwarzanie danych.

3. Dostępność dla osób z niepełnosprawnościami

- Interfejs powinien spełniać standardy dostępności WCAG 2.1 AA.
- System powinien być kompatybilny z technologiami wspomagającymi używanymi przez osoby z niepełnosprawnościami.

3.5 Słownik

Ograniczenia

Priorytet wysoki

Zapewnienie danych zgodnie ze standardami dostępności

- Wszystkie dane muszą być dostarczane w sposób zgodny ze standardami dostępności (np. WCAG 2.1 AA).

Prezentacja wyników w różnych formatach

- System musi umożliwiać eksport danych w co najmniej trzech formatach: CSV, JSON i XML.

Integracja z CKAN

- System musi być kompatybilny z platformą CKAN i umożliwiać integrację poprzez API.

Kompatybilność systemu

- Minimalne wymagania dla urządzeń mobilnych: Android 10 i iOS 12.

Wsparcie dla Internet Explorer 11

- System musi zapewniać podstawową funkcjonalność w przeglądarce Internet Explorer 11.

Infrastruktura na terenie Unii Europejskiej

- Wszystkie serwery i centra danych muszą znajdować się na terytorium UE.

Zgodność z RODO i NIS2

- System musi spełniać wymagania **RODO** (Rozporządzenie o Ochronie Danych Osobowych) oraz **NIS2** (Dyrektywa w sprawie bezpieczeństwa sieci i informacji).

Priorytet średni

Ograniczenie dostępu do panelu administracyjnego

- Panel administracyjny musi być dostępny wyłącznie z wewnętrznej sieci (intranet).

Priorytet niski

Dostępność dla osób z niepełnosprawnościami

- Interfejs systemu musi być zgodny ze standardami WCAG 2.1 AA i obsługiwać technologie wspomagające.

3.6 Słownik

Słownik

Administracja publiczna: Instytucje rządowe i samorządowe tworzące i udostępniające dane publiczne.

Aktualizacja danych: Proces regularnego sprawdzania dostępności nowych danych, ich pobierania, walidacji i zapisywania w systemie, aby utrzymać bazę danych w stanie aktualności i zgodności z rzeczywistością. System powinien umożliwiać ciągłą aktualizację danych i automatyczne udostępnianie zaktualizowanych danych.

Analiza danych: Proces przetwarzania i badania danych w celu identyfikacji trendów, wzorców oraz anomalii.

Analiza ruchu: Badanie danych dotyczących aktywności użytkowników w systemie.

API (Interfejs Programowania Aplikacji): Interfejs umożliwiający komunikację i wymianę danych między różnymi systemami lub aplikacjami. System ma udostępniać dane poprzez API.

Archiwizacja danych: Proces przechowywania danych historycznych w celu ich zabezpieczenia i możliwości odtworzenia.

Autentyczność danych: Zapewnienie, że dane są wiarygodne i pochodzą z zaufanego źródła. System powinien zapewniać autentyczność danych.

Autoryzacja: Proces weryfikacji, czy użytkownik ma uprawnienia do wykonania określonej akcji lub dostępu do określonych zasobów.

Automatyzacja: Wykorzystanie systemów do wykonywania procesów biznesowych w celu minimalizacji błędów ludzkich i zwiększenia efektywności.

Awarie: Nieprawidłowe działanie systemu. System powinien monitorować swój stan w celu wykrywania awarii i informować o nich administratora.

Backup danych: Kopia zapasowa danych, tworzona w celu możliwości przywrócenia systemu do poprzedniego stanu w przypadku awarii lub błędów.

Bezpieczeństwo danych: Ochrona danych przed nieautoryzowanym dostępem, utratą lub uszkodzeniem. System ma zapewniać bezpieczeństwo danych i chronić przed atakami.

Błędy danych: Niepoprawne, niekompletne lub błędne informacje znajdujące się w systemie. Użytkownicy powinni mieć możliwość zgłaszania błędów danych.

CKAN: Popularna platforma do zarządzania i udostępniania otwartych danych. System ma być zintegrowany z CKAN i zgodny ze standardami CKAN.

Dane otwarte: Dane publiczne udostępniane w sposób otwarty, zgodnie z określonymi standardami, umożliwiające ich swobodne wykorzystywanie i ponowne użycie.

Dane publiczne: Dane tworzone i udostępniane przez administrację publiczną. Przykłady to rozkłady jazdy, dane o położeniu pojazdów, stężenia zanieczyszczeń, położenia toalet miejskich, godziny dostępności urzędów.

DCAT: Standard metadanych opisujący publiczne dane. System powinien walidować dane pod kątem zgodności ze standardem DCAT i być zgodny z nim.

Dostęp do danych: Możliwość korzystania z danych przez uprawnionych użytkowników. System ma ułatwiać dostęp do różnych źródeł danych dla klientów i zapewniać stały oraz ciągły dostęp do danych.

Dystrybucje zbiorów danych: Konkretnie formy udostępniania zbiorów danych (np. pliki w określonym formacie, dostęp przez API). System ma zarządzać dystrybucjami zbiorów danych.

Efektywność: Zdolność systemu do sprawnego i szybkiego działania, np. minimalizowanie czasu potrzebnego do wyszukiwania danych.

Filtrowanie danych: Proces wybierania danych spełniających określone kryteria. Użytkownicy mają mieć możliwość filtrowania zbiorów danych.

Formaty danych: Sposoby zapisu i organizacji danych (np. JSON, XML, CSV, RDF). System ma udostępniać dane w różnych formatach oraz umożliwiać import i eksport danych w różnych formatach.

Generowanie raportów: Automatyczne tworzenie podsumowań i wizualizacji danych. System ma generować raporty o ruchu.

Historia danych: Zapis zmian danych w czasie, umożliwiający śledzenie ich ewolucji. System powinien utrzymywać historię danych i przedstawiać historię zmian w przystępny sposób.

Integracja z CKAN: Połączenie i współpraca systemu z platformą CKAN.

Interfejs użytkownika (UI): Element systemu umożliwiający interakcję użytkownika z systemem. Interfejs powinien być przejrzysty i adaptowalny do rozmiaru ekranu.

Intranet: Prywatna sieć komputerowa, dostępna tylko dla pracowników organizacji. Dostęp do panelu administracyjnego powinien być ograniczony do intranetu.

Jakość danych: Stopień, w jakim dane są dokładne, kompletne, spójne i aktualne. System ma ulepszać jakość danych i monitorować ją.

Logi systemu: Zapis aktywności systemu, przechowywany w celu audytu i śledzenia błędów.

Metadane: Dane opisujące inne dane, zawierające informacje o ich pochodzeniu, formacie, strukturze itp.. System ma zarządzać metadanymi udostępnionych zbiorów danych.

Monitorowanie systemu: Ciągłe sprawdzanie działania systemu w celu wykrywania problemów, anomalii lub zagrożeń.

Oceny użytkowników: Opinie i oceny zbiorów danych wystawiane przez użytkowników. System ma zbierać i analizować oceny użytkowników oraz prezentować oceny danych.

Otwartość danych (piąty poziom): Zgodność z zasadami otwartych danych, w tym m.in. z formatem RDF (Resource Description Framework), umożliwiającym łączenie danych.

Panel administracyjny: Interfejs służący do zarządzania systemem, dostępny zazwyczaj dla administratorów.

Procesy biznesowe: Zestaw powiązanych działań lub zadań, które razem realizują określony cel biznesowy. Źródła opisują różne procesy biznesowe związane z zarządzaniem danymi.

Przeglądanie danych: Możliwość oglądania i zapoznawania się z zawartością zbiorów danych.

Przejrzystość interfejsu: Zrozumiałość i łatwość obsługi interfejsu użytkownika.

RDF (Resource Description Framework): Standard formatu danych używany w semantycznym Webie, umożliwiający powiązanie różnych zbiorów danych. Zgodność z RDF jest wymogiem piątego poziomu otwartości danych.

Rejestracja nowych źródeł danych: Proces dodawania informacji o nowych źródłach danych do systemu. Może odbywać się poprzez API lub formularz.

Retencja danych: Okres przechowywania danych w systemie.

RODO (Ogólne Rozporządzenie o Ochronie Danych): Przepisy prawne dotyczące ochrony danych osobowych. System powinien być zgodny z RODO.

Rola użytkownika: Określona kategoria użytkowników systemu z przypisanymi uprawnieniami dostępu.

Skalowalność systemu: Zdolność systemu do obsługi rosnącej liczby użytkowników i danych bez utraty wydajności.

SLA (Service Level Agreement): Umowa dotycząca poziomu świadczonych usług, np. dostępności systemu. Źródła wspominają o SLA na poziomie 99.99999%.

Sprawdzanie poziomu dostępu: Weryfikacja uprawnień użytkownika do określonych danych lub funkcji systemu.

Standardy danych: Ustalone wytyczne dotyczące formatu, struktury i jakości danych. Dane muszą spełniać określone standardy, aby móc je udostępniać.

Statystyki wyszukiwań: Dane dotyczące zapytań wyszukiwania użytkowników.

System zarządzania otwartymi danymi publicznymi: System informatyczny mający na celu sprawne zarządzanie danymi publicznymi i ich udostępnianie w sposób otwarty.

Szyfrowanie danych: Kodowanie danych w celu ochrony ich poufności. Dotyczy zarówno danych wrażliwych, jak i przesyłu danych.

Śledzenie historii zmian: Rejestrowanie modyfikacji danych w czasie.

Udostępnianie danych: Udostępnianie zbiorów danych użytkownikom w różnych formatach i poprzez różne kanały (np. aplikacja webowa, API).

Uwierzytelnianie: Proces weryfikacji tożsamości użytkownika.

Walidacja danych: Sprawdzanie poprawności danych pod kątem zgodności z określonymi regułami i standardami.

Wersjonowanie danych: Zarządzanie różnymi wersjami zbiorów danych i dystrybucji.

Wyszukiwanie danych: Proces znajdowania danych spełniających określone kryteria. System ma minimalizować czas potrzebny do wyszukiwania danych i prezentować wyniki wyszukiwania w różnych formatach.

Zabezpieczenie przed atakami: Działania mające na celu ochronę systemu przed cyberatakami i nieautoryzowanym dostępem.

Zgodność z RODO i NIS2: Przestrzeganie przepisów dotyczących ochrony danych osobowych (RODO) oraz bezpieczeństwa systemów sieciowych i informacyjnych (NIS2).

Zgłaszanie błędów: Proces umożliwiający użytkownikom informowanie o nieprawidłowościach w danych.

Zbiory danych: Kolekcje danych udostępnianych przez system. System ma umożliwiać wygodne wprowadzanie, przeglądanie, filtrowanie i wyszukiwanie zbiorów danych oraz zarządzanie nimi.

4. Wymagania użytkownika

4.1 Wymagania funkcjonalne

4.1.1 Zarządzanie zbiorami danych



UC0001: Wyświetlanie zbiorów danych

Autor: Marharyta Minich

Scenariusz Główny

PRE: Użytkownik na stronie głównej

1. Użytkownik wybiera opcję "Lista Zbiorów Danych".
2. System pobiera z bazy zbiorów danych dane.

[dane pobrane poprawnie]

3. System wyświetla listę zbiorów danych.

final: *success*

POST: Na ekranie wyświetla się lista zbiorów danych.

Scenariusz Alternatywny 1

1-2. Jak w scenariuszu głównym.

[brak danych]

2a. System wyświetla komunikat: "Brak dostępnych zbiorów danych".

final: *failure*

POST: Na ekranie wyświetla się komunikat informujący o braku danych.

Scenariusz Alternatywny 2

1-2. Jak w scenariuszu głównym.

[dane pobrane niepoprawnie]

3b. System wykrywa awarię bazy danych.

4b. System wyświetla komunikat o błędzie.

final: *failure*

POST: Na ekranie wyświetla się komunikat informujący o błędzie.



UC0002: Wyszukiwanie zbiorów danych

Autor: Ivan Rusinovich

SCENARIUSZ GŁÓWNY

PRE: Klient/Użytkownik danych znajduje się na stronie głównej

1. Klient/Użytkownik wybiera opcję "Wyszukaj zbiór danych"

- 2. Klient/Użytkownik danych wpisuje frazę wyszukiwania lub wybiera kryteria (tematyka, lokalizacja, data publikacji)
- 3. System przetwarza zapytanie i aktywuje mechanizm wyszukiwania
- 4. System pobiera pasujące zbiory danych z bazy danych
- 5. System generuje listę wyników wyszukiwania zawierającą metadane (tytuł, opis, data publikacji, format)
- 6. System wyświetla listę wyników z opcjami sortowania i filtrowania

final: success

POST: Klient/Użytkownik danych widzi listę pasujących zbiorów danych z możliwością dalszej analizy

SCENARIUSZ ALTERNATYWNY 1 – Brak wyników wyszukiwania

- 1-2. tak jak w SCENARIUSZU GŁÓWNYM
- [brak wyników]
- 3a. System nie znajduje żadnych pasujących zbiorów danych
- 4a. System wyświetla komunikat informacyjny o braku wyników i proponuje zmianę kryteriów

final: failure

POST: Klient/Użytkownik danych nie otrzymuje wyników, ale może zmodyfikować zapytanie

SCENARIUSZ ALTERNATYWNY 2 – Błąd przetwarzania zapytania

- 1-2. tak jak w SCENARIUSZU GŁÓWNYM
- [błąd systemu]
- 3b. System napotyka błąd podczas przetwarzania zapytania
- 4b. System wyświetla komunikat o błędzie technicznym
- 5b. Klient/Użytkownik danych może spróbować ponownie później lub skontaktować się z pomocą techniczną

final: failure

POST: Wyniki nie zostają wyświetlone; użytkownik informowany jest o błędzie przetwarzania

zbiór danych
tytuł: string
opis: string
data_publicacji: date
tematyka: string
lokalizacja: string
format: string

kryteria wyszukiwania
fraza: string
tematyka: string
lokalizacja: string
data_od: date
data_do: date

interfejs wyszukiwania
pole_frazy: string
filtr_tematyki: list
filtr_lokalizacji: list
zakres_dat: date range
przycisk_szukaj: bool
opcje_sortowania: list

UC0003: Filtrowanie zbiorów danych

Autor: Wiktoria Jeleniewicz

Scenariusz Główny

PRE: Użytkownik jest zalogowany. Wyświetlona jest lista zbiorów danych.

1. Użytkownik wprowadza kryteria filtrowania.

[baza danych jest dostępna]

2. System filtruje zbiory danych.

[znaleziono zbiory danych]

3. System wyświetla listę zbiorów danych.

final: *success*

POST: Na ekranie wyświetla się lista zbiorów danych spełniających wprowadzone kryteria filtrowania.

Scenariusz Alternatywny 1

1–2. Jak w scenariuszu głównym.

[nie znaleziono zbiorów danych]

3a. System wyświetla komunikat: "Brak wyników. Spróbuj zmienić kryteria wyszukiwania."

final: *failure*

POST: Na ekranie wyświetla się komunikat informujący o niepowodzeniu.

Scenariusz Alternatywny 2

1–1. Jak w scenariuszu głównym.

[baza danych jest niedostępna]

2b. System wyświetla komunikat: "Niepowodzenie. Spróbuj ponownie później."

final: *failure*

POST: Na ekranie wyświetla się komunikat informujący o niepowodzeniu.

kryteria filtrowania
dataOd: data
dataDo: data
autor: tekst
tematDanych: tekst

komunikat
tekst: tekst

zbiór danych
nazwa: tekst
opis: tekst
dataModyfikacji: data
autor: tekst
tematDanych: tekst

UC0005: Prezentacja historii zmian zbioru danych

Autor: Kamil Dobkowski

Scenariusz Główny:

PRE: Użytkownik jest zalogowany. Wyświetlona jest lista zbiorów danych.

- 1. **Użytkownik** wybiera opcję "wyświetl historie zmian".
[Istnieje historia zmian] AND [Połączenie z bazą poprawne]
 - 2. **System** wyświetla historie zmian zbioru danych
- POST:** System wyświetlił na ekranie historie zmian zbioru danych.

Scenariusz Alternatywny 1 - Brak historii zmian zbioru danych

- 1 Jak w scenariuszu głównym
- [Brak historii zmian]**
- 2a. **System** wyświetla komunikat o braku historii zmian
- final:** *failure*
- POST:** System nie wyświetlił historii zmian zbioru danych

Scenariusz Alternatywny 2 - Błąd bazy danych

- 1 Jak w scenariuszu głównym
- [Błąd bazy danych]**
- 2b. **System** wyświetla komunikat o niedostępności bazy danych
- final:** *failure*
- POST:** System nie wyświetlił historii zmian zbioru danych



UC0007: Edycja zbioru danych

Autor: Grzegorz Sawicki

Scenariusz Główny UC0007

PRE: Dostawca danych jest zalogowany, posiada uprawnienia do edycji zbioru. Dostawca na stronie głównej (dashboard).

1. danych wybrał opcję edycji zbioru z listy zbiorów
2. System wyświetla formularz z obecnymi informacjami o zbiorze.
3. Dostawca danych wprowadza zmiany w formularzu.
- <<invoke>> UC0008: Walidacja danych
- [wynik walidacji poprawny]
4. System sprawdza dostępność bazy danych.
- [baza danych dostępna]
5. System zapisuje zaktualizowany zbiór danych w bazie.
6. System wyświetla komunikat potwierdzający pomyślną edycję zbioru danych.
- final: success

POST: Dany zbiór zawiera zaktualizowane dane w bazie, zmiana jest uwzględniona w historii zmian.

****Scenariusz Alternatywny UC0007-A**

- 1.-3. jak w scenariuszu głównym UC0007.
- [wynik walidacji niepoprawny]
- 4a. System wyświetla komunikat o błędzie walidacji.
- final: failure

POST: Dany zbiór pozostaje bez zmian.

****Scenariusz Alternatywny UC0007-B**

- 1.-4. jak w scenariuszu głównym UC0007.
- [baza danych niedostępna]
- 5b. System wyświetla komunikat o niedostępności bazy danych.
- final: failure

POST: Dany zbiór pozostaje bez zmian.



UC0009: Importowanie danych

Autor: Ilya Rohoutseu

Scenariusz Główny

PRE: Dostawca danych ma dostęp do systemu i jest na stronie głównej **1.** Administrator wybiera opcję "Importowanie danych"

2. System wyświetla interfejs do przesłania pliku z danymi.

3. Dostawca danych przesyła plik z danymi.

[Obsługiwany format pliku]

4. System waliduje przesłane dane.

[Dane poprawne]

5. System zapisuje nowy zbiór danych w bazie i rejestruje go w katalogu zbiorów.

[Dane poprawnie zapisane]

6. System wyświetla komunikat "Pomyślne zaimportowanie nowego zbioru danych".

final: *success*

POST: Dane zostały importowane do systemu.

Scenariusz Alternatywny 1

1–3. Jak w scenariuszu głównym.

[Nieobsługiwany format pliku]

4a. System wyświetla komunikat informujący o błędzie formatu pliku.

final: *failure*

POST: Dane nie zostały importowane do systemu.

Scenariusz Alternatywny 2

1–5. Jak w scenariuszu głównym.

[Dane niepoprawne]

6b. System wyświetla komunikat o błędach.

final: *failure*

POST: Dane nie zostały importowane do systemu.

Słownik

Plik

- **Nazwa:** tekst
- **Format:** tekst
- **Rozmiar:** liczba

Zbiór danych

- **Nazwa:** tekst
- **Rozmiar:** liczba
- **Data:** data

Komunikat

- **Nazwa:** tekst
- **Treść:** tekst

UC0011: Eksportowanie danych w różnych formatach

Autor: Aleksander Piluk

Aktorzy: Użytkownik danych, System

Scenariusz Główny

PRE: Użytkownik danych ma dostęp do systemu i wyświetla listę zbiorów danych.

1. Użytkownik danych wybiera opcję eksportu danych.
2. System sprawdza istnienie danych do eksportu.

[Dane do eksportu istnieją]

3. System wyświetla formularz wyboru formatu eksportu danych.
4. Użytkownik danych wybiera format eksportu danych.
5. System generuje plik w wybranym formacie eksportu danych.

[Dane wygenerowane poprawnie]

6. System wyświetla okno pobierania wygenerowanego pliku.
7. Użytkownik danych wybiera opcję pobierz.
[Pobieranie zakończone sukcesem]
8. System wyświetla komunikat o poprawnym pobraniu pliku.
9. Użytkownik danych wybiera opcję zamknij.

final: *success*

POST: Eksport danych poprawny.

Scenariusz Alternatywny 1 - Brak danych do eksportu

1–2. Jak w scenariuszu głównym.

[Brak danych do eksportu]

- 3a. System wyświetla komunikat o braku danych do eksportu.
- 4a. Użytkownik danych wybiera opcję zamknij.

final: *failure*

POST: Brak danych do eksportu.

Scenariusz Alternatywny 2 - Błąd generowania

1–5. Jak w scenariuszu głównym.

[Błąd generowania danych]

- 6b. System wyświetla komunikat o błędzie generowania danych.
- 7b. Użytkownik danych wybiera opcję zamknij.

final: *failure*

POST: Błąd generowania danych.

Scenariusz Alternatywny 3 - Błąd pobierania

1–7. Jak w scenariuszu głównym.

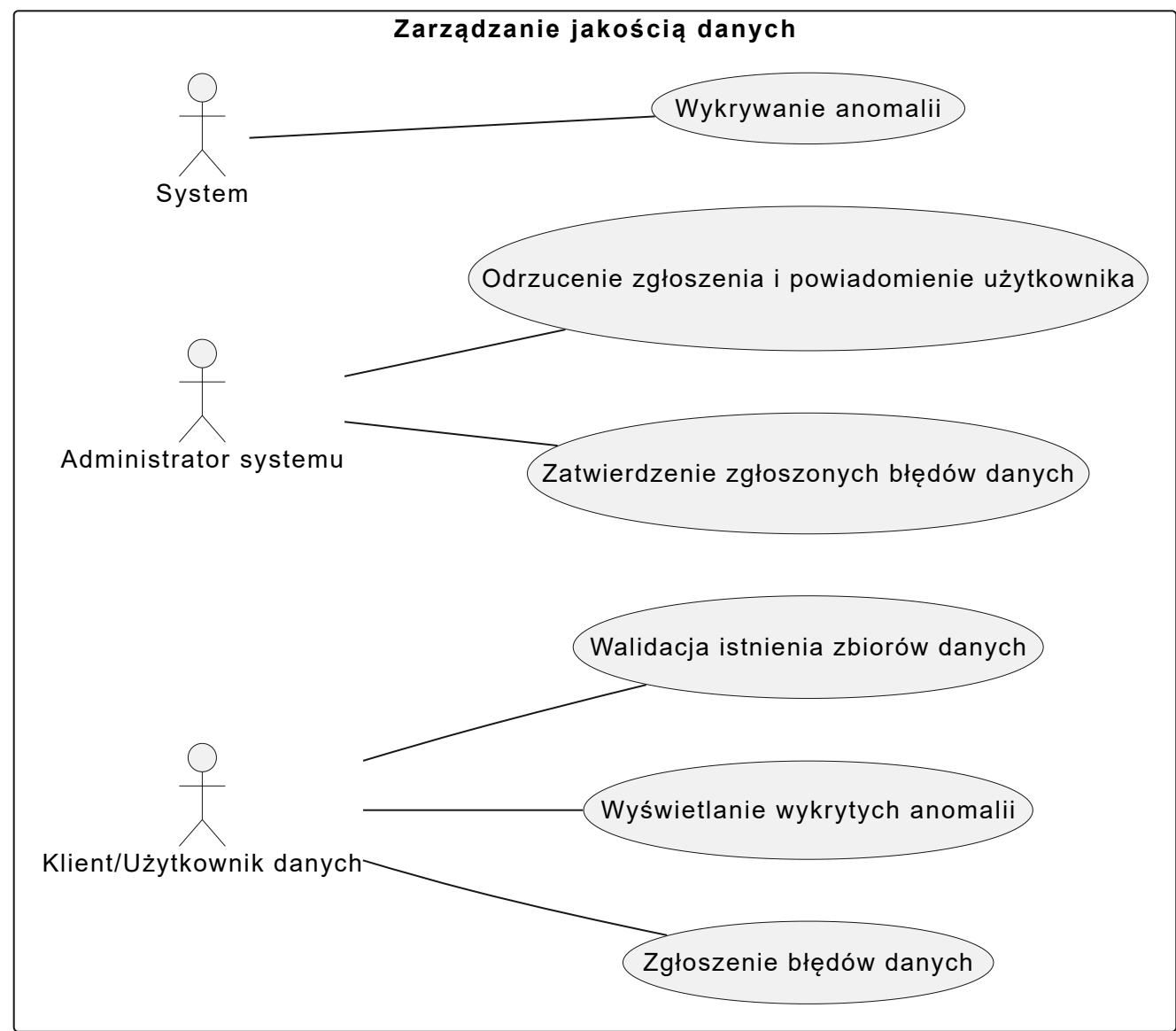
[Błąd pobierania]

- 8c. System wyświetla komunikat o błędzie pobierania.
- 9c. Użytkownik danych wybiera opcję zamknij.
- final: failure
- POST: Błąd pobierania danych.

Format danych eksportu	Plik eksportu danych
Format: tekst	Plik: struktura

Dane do eksportu
Dane: struktura

4.1.2 Zarządzanie jakością danych



UC2001: Zgłoszenie błędów danych

Autor: Grzegorz Sawicki

Scenariusz Główny UC2001

PRE: Użytkownik jest zalogowany, użytkownik wybrał opcję zgłoszenia błędu danych.

- 1. System wyświetla formularz zgłoszenia błędu.
- 2. Użytkownik wypełnia formularz.

[baza danych ok]

[standardowe przywileje zgłaszania błędów]

- 3. System zapisuje zgłoszenie.
- 4. System wyświetla komunikat potwierdzający.

final: *success*

POST: Zgłoszenie jest zapisane do dalszej analizy przez administratora.

Scenariusz Alternatywny UC2001-A

- 1.-2. jak w scenariuszu głównym UC2001.

[baza danych niedostępna]

- 3a. System wyświetla komunikat o błędzie bazy danych.

final: *failure*

POST: brak zmian, użytkownik wraca do okna danego zbioru.

Scenariusz Alternatywny UC2001-B

- 1.-2. jak w scenariuszu głównym UC2001.

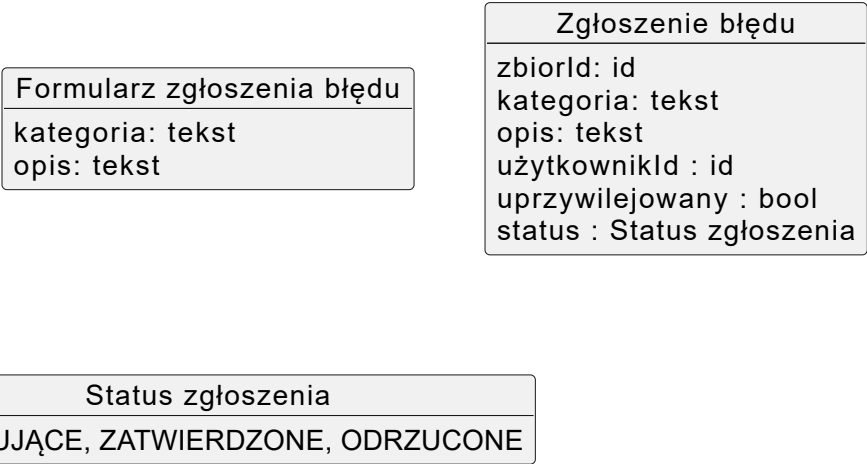
[baza danych ok]

[wyższe przywileje zgłaszania błędów danych]

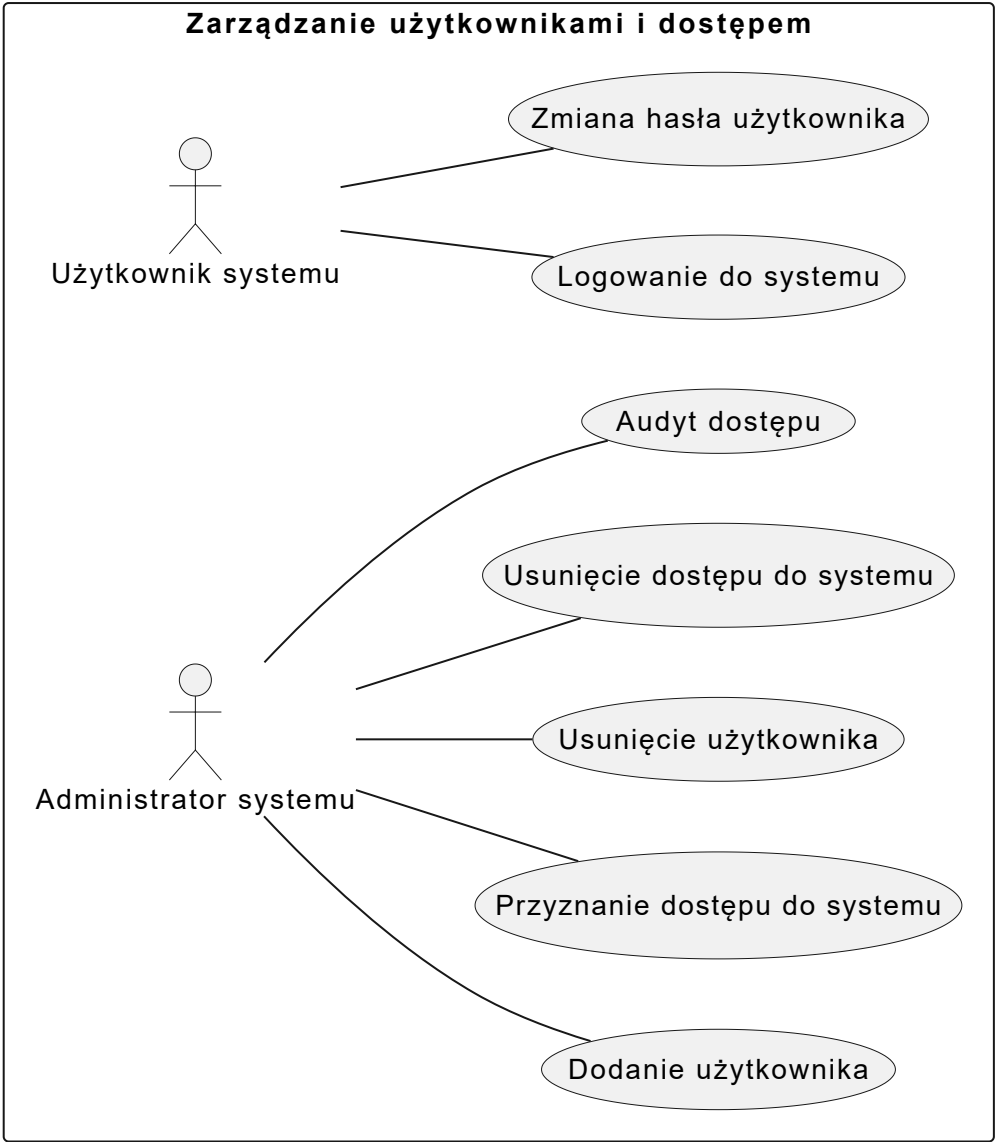
- 3b. System zapisuje zgłoszenie z informacją o wyższej wiarygodności.
- 4b. System wyświetla komunikat potwierdzający.

final: *success*

POST: Zgłoszenie jest zapisane z większym priorytetem do dalszej analizy przez administratora.



4.1.3 Zarządzanie użytkownikami i dostępem



UC0021: Logowanie do systemu

Autor: Albert Grochowski

Scenariusz Główny

PRE: Użytkownik znajduje się na stronie głównej systemu i jest niezalogowany.

1. Użytkownik wybiera opcję Zaloguj.
2. System wyświetla formularz logowania.
3. Użytkownik wprowadza login i hasło.
4. System weryfikuje login i hasło.

[Dane poprawne]

5. System loguje użytkownika.
6. System wyświetla potwierdzenie zalogowania.

final: success

POST: Użytkownik jest zalogowany.

Scenariusz Alternatywny 1

1–4. Jak w scenariuszu głównym.

[Nieprawidłowy login lub hasło]

5a. System wyświetla informację o błędzie logowania.

6a. Użytkownik wybiera opcję "Ponów logowanie".

Powrót do kroku 2 w scenariuszu głównym.

Scenariusz Alternatywny 2

1–5a. Jak w scenariuszu alteratywnym 1.

[anulowanie logowania]

6b. Użytkownik wybiera opcję "anuluj logowanie".

7b. System zamyka formularz logowania.

final: *failure*

POST: Użytkownik nie zalogowany.

formularz logowania

login: tekst
hasło: tekst

UC0022: zmiana hasła

Autor: Albert Grochowski

Scenariusz Główny

PRE: Użytkownik znajduje się na stronie głównej systemu

1. Użytkownik wybiera opcję "zmień hasło".

2. System wyświetla formularz zmiany hasła.

3. Użytkownik wprowadza dane zmiany hasła.

4. System weryfikuje dane zmiany hasła.

[Dane poprawne]

5. System zmienia hasło.

6. System wyświetla potwierdzenie zmiany hasła.

final: *success*

POST: Użytkownik jest zalogowany, hasło zmienione.

Scenariusz Alternatywny 1

1–4. Jak w scenariuszu głównym.

[Dane niepoprawne]

5a. System wyświetla informację o niepoprawnych danych.

6a. Użytkownik wybiera opcję "Ponów".

Powrót do kroku 2 w scenariuszu głównym.

Scenariusz Alternatywny 2

1–4. Jak w scenariuszu głównym.

[błąd bazy danych]

5b. System wyświetla komunikat o błędzie bazy danych.

6b. użytkownik wybiera opcję "zamknij".

7b. System zamyka formularz zmiany hasła.

final: *failure*

POST: Użytkownik bez zmiany hasła.

hasła
stare hasło: tekst
nowe hasło: tekst

UC0023: Dodanie Użytkownika

Autor: Kamil Dobkowski

Scenariusz Główny:

PRE: Administrator systemu jest zalogowany i posiada odpowiednie uprawnienia do zarządzania użytkownikami, jest na stronie głównej.

1. **Administrator systemu** wybiera opcję "dodaj nowego użytkownika".
2. **Administrator systemu** wprowadza dane użytkownika.
3. **System** weryfikuje poprawność danych **[Dane poprawne]**
4. **System zapisuje dane w bazie danych [Zapis w bazie udany]**
5. **System** wyświetla komunikat o poprawnym utworzeniu użytkownika

final: *success*

POST: System zawiera nowe konto użytkownika w bazie danych.

Scenariusz Alternatywny 1 - Niepoprawne dane

1-3. Jak w scenariuszu głównym

[Niepoprawne dane]

4a. **System** wyświetla komunikat o niepoprawnych danych

Powrót do 2. w scenariuszu głównym

Scenariusz Alternatywny 2 - Błąd bazy danych

1-5. Jak w scenariuszu głównym

[Błąd bazy danych]

6a. **System** wyświetla komunikat o niedostępności bazy danych

final: *failure*

POST: Nie dodano użytkownika.

Dane użytkownika
Imię: tekst
Nazwisko: tekst
Login: tekst
Hasło: tekst

UC0024: Przyznanie dostępu do systemu

Autor: Kiril Horobets

Scenariusz Główny

PRE: Administrator systemu jest zalogowany do systemu i jest na stronie głównej.

1. Administrator wybiera użytkownika, któremu chce przyznać dostęp.
2. Administrator określa poziom dostępu.
3. Administrator zatwierdza zmiany.
4. System weryfikuje poprawność wprowadzonych danych.

[Dane poprawne]

5. System zapisuje zmiany w bazie zbiorów danych.
6. System wyświetla komunikat o pomyślnym przyznaniu dostępu.

final: *success*

POST: Użytkownik posiada nowy poziom dostępu zapisany w systemie.

Scenariusz Alternatywny 1

1–4. Jak w scenariuszu głównym.

[Dane niepoprawne]

- 4a. System wyświetla komunikat o błędzie i prosi administratora o poprawienie danych.
- 5a. Administrator wybiera opcję „Anuluj”.
- 6a. System anuluje operację przyznania dostępu.

final: *failure*

POST: Użytkownik nie posiada nowego poziomu dostępu.

Scenariusz Alternatywny 2

1–4. Jak w scenariuszu głównym.

[Dane niepoprawne]

- 5b. System wykrywa, że użytkownik już posiada przyznany dostęp na wybranym poziomie.
- 6b. System wyświetla komunikat informujący o istniejącym dostępie.
- 7b. Administrator wybiera opcję „Anuluj”.
- 8b. System anuluje operację przyznania dostępu.

final: *failure*

POST: Brak zmian w poziomach dostępu użytkownika.

Słownik

Poziom dostępu

- **Nazwa:** tekst
- **Zakres:** lista
- **Data:** data

Baza zbiorów danych

- **Nazwa:** tekst
- **Rozmiar:** liczba
- **Data:** data

Komunikat

- **Nazwa:** tekst
- **Treść:** tekst
- **Załącznik:** plik
- **Data:** data
- **Ważność:** tekst

UC0025: Usunięcie użytkownika

Autor: Wiktoria Jeleniewicz

Scenariusz Główny

PRE: Administrator jest zalogowany. Wyświetlone jest menu główne.

1. Administrator wybiera opcję "Usuń użytkownika".
2. System wyświetla formularz danych użytkownika.
3. Administrator wprowadza dane użytkownika.
4. Administrator wybiera opcję "Usuń".
5. System waliduje dane użytkownika.

[dane poprawne]

6. System usuwa użytkownika.
7. System wyświetla komunikat "Usunięto użytkownika".

final: *success*

POST: Użytkownik nie znajduje się w bazie użytkowników.

Scenariusz Alternatywny 1

1–5. Jak w scenariuszu głównym.

[dane niepoprawne]

- 6a. System wyświetla komunikat "Nie znaleziono użytkownika o podanych danych".
- 7a. Administrator wybiera opcję "Zamknij".
- 8a. System zamyka formularz danych użytkownika.

final: *failure*

POST: Na ekranie wyświetla się komunikat informujący o niepowodzeniu.

Scenariusz Alternatywny 2

1–5. Jak w scenariuszu głównym.

[baza danych nie odpowiada]

6b. System wyświetla komunikat: "Niepowodzenie. Spróbuj ponownie później."

7b. Administrator wybiera opcję "Zamknij".

8b. System zamyka formularz danych użytkownika.

final: *failure*

Użytkownik	Dane użytkownika
id: number login: tekst hasło: tekst email: tekst	id: number email: tekst

UC0026: Modyfikacja dostępu do systemu

Autor: Marcin Marszewski

Scenariusz Główny - UC0026

PRE: Administrator systemu jest zalogowany do systemu i posiada uprawnienia zmiany dostępu, jest na stronie głównej.

1. Administrator wybiera opcję zmiany dostępu użytkownikowi.

2. System wyświetla listę użytkowników.

3. Administrator wybiera użytkownika.

4. System wyświetla widok określania dostępu.

5. Administrator określa poziom dostępu.

6. Administrator zatwierdza zmiany.

7. System weryfikuje poprawność wprowadzonych danych.

[Dane poprawne]

8. System zapisuje zmiany w bazie danych.

9. System wyświetla komunikat o pomyślnej zmianie dostępu.

final: *success*

POST: Użytkownik posiada nowy poziom dostępu zapisany w systemie.

Scenariusz Alternatywny 1 - UC0026

1–7. Jak w scenariuszu głównym.

[Dane niepoprawne]

8a. System wyświetla komunikat o błędzie danych.

9a. Administrator wybiera opcję „Ponów”.

Powrót do zdania 4 w Scenariuszu Głównym - UC0026

Scenariusz Alternatywny 2 - UC0026

1–7. Jak w scenariuszu głównym.

[Błąd bazy danych]

8a. System wyświetla komunikat o błędzie bazy.

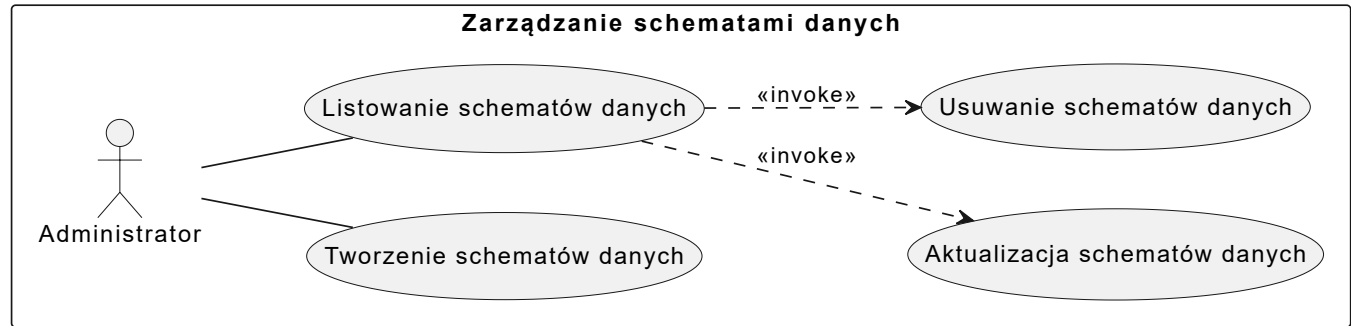
9a. Administrator wybiera opcję „Anuluj”.

final: *failure*

POST: Poziom dostępu użytkownika w systemie się nie zmienił.

Użytkownik
id: tekst
imie : tekst
nazwisko : tekst
login : tekst
hasło : tekst
dostęp : enum[]

4.1.4 Zarządzanie schematami danych



UC0028: Tworzenie schematów danych

Autor: Kamil Dobkowski

Scenariusz Główny

PRE: Administrator ma uprawnienia do dodawania schematów danych, jest na stronie głównej.

1. Administrator wybiera opcję tworzenia nowego schematu danych.
2. System wyświetla formularz dodawania schematu.
3. Administrator wprowadza schemat danych.
4. System waliduje schemat.

[Schemat poprawny]

5. System synchronizuje schemat z CKAN.

[Synchronizacja z CKAN poprawna]

6. System zapisuje schemat do bazy danych.
7. System wyświetla komunikat o dodaniu schematu. **final:** *success*

POST: Schemat jest dodany do bazy danych.

Scenariusz Alternatywny 1

1–4. Jak w scenariuszu głównym.

[Schemat niepoprawny]

4a. System wyświetla komunikat o błędnym schemacie.

Powrót do kroku 3. w scenariuszu głównym.

Scenariusz Alternatywny 2

1–5. Jak w scenariuszu głównym.

[Synchronizacja z CKAN nieudana]

6a. System wyświetla komunikat o błędzie synchronizacji z CKAN

final: *failure*

POST: Nie dodano schematu.

Słownik

Schemat

- **Nazwa:** tekst
 - **Format:** enum
 - **Wartość:** raw
-

UC0030: Listowanie schematów danych

Autor: Marharyta Minich

Scenariusz Główny

PRE: Użytkownik jest zalogowany. Wyświetlone jest menu główne.

1. Użytkownik wybiera opcję "Pokaż schematy danych"

[dane pobrane poprawnie]

2. System wyświetla listę schematów

final: *success*

POST: Lista zbiorów została wyświetlona użytkownikowi

Scenariusz Alternatywny 1

1-1. Jak w scenariuszu głównym.

[brak połączenia]

2a. System wyświetla komunikat: "Błąd podczas połączenia do bazy danych".

final: *failure*

POST: Na ekranie wyświetla się komunikat informujący o błędzie

Scenariusz Alternatywny 2

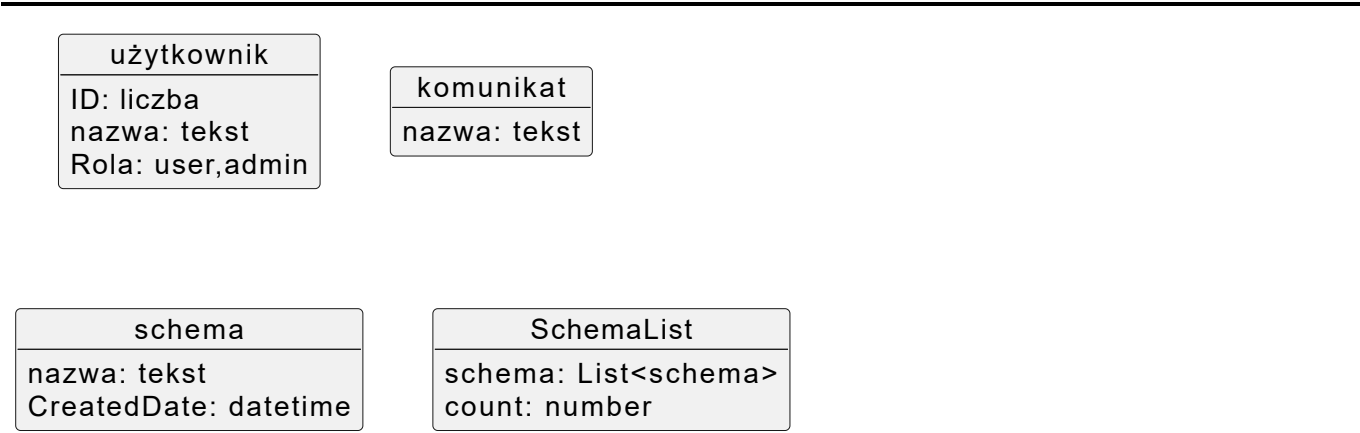
1-2. Jak w scenariuszu głównym.

[brak uprawnień]

3b. System wyświetla komunikat: "Brak uprawnień do przeglądania schematów".

final: failure

POST: Na ekranie wyświetla się komunikat informujący o błędzie.



UC0031: Usuwanie schematu danych

Autor: Kiril Horobets

Scenariusz Główny

PRE: Administrator jest zalogowany i znajduje się w Lista Schematow Danych.

1. Administrator wybiera schemat danych do usunięcia.
2. Administrator zatwierdza usunięcie.
3. System weryfikuje poprawność operacji.
4. System usuwa schemat danych z bazy danych.
5. System synchronicznie usuwa schemat z platformą CKAN.
6. System wyświetla komunikat o pomyślnym usunięciu schematu danych.

final: success

POST: Schemat danych wybrany przez Administratora został pomyślnie usunięty z bazy danych i operacja została zsynchronizowana z platformą CKAN.

Scenariusz Alternatywny 1

1–3. Jak w scenariuszu głównym.

[Dane niepoprawne]

- 4a. System wykrywa, że schemat danych jest powiązany z istniejącym zbiorem danych.
- 5a. System wyświetla komunikat o błędzie i uniemożliwia usunięcie schematu.
- 6a. Anulowanie usunięcia.

final: failure

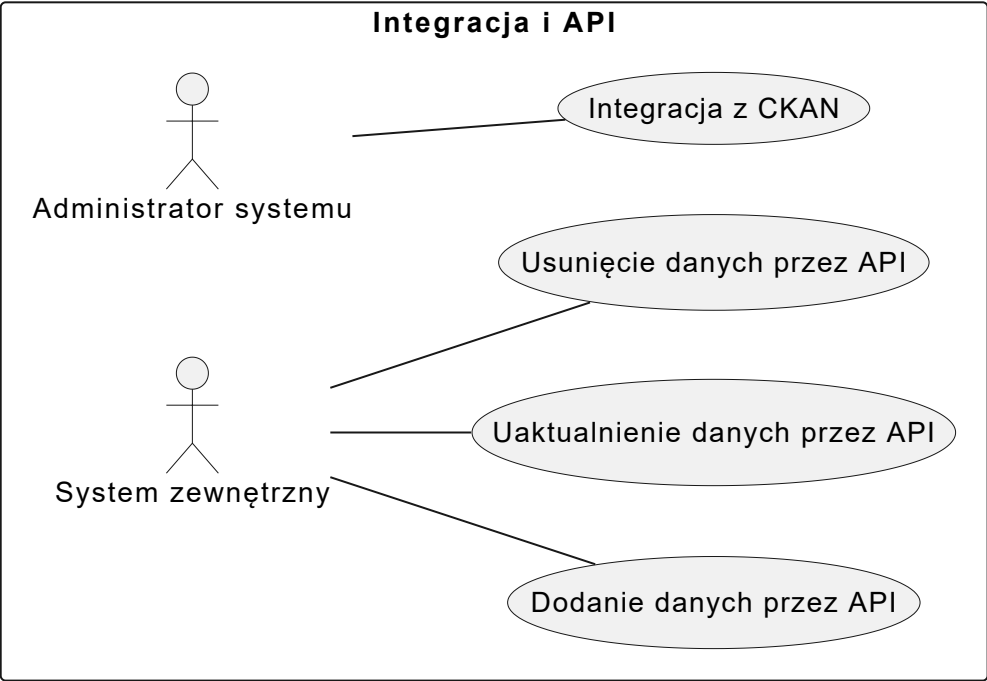
POST: Schemat danych nie został usunięty.

Scenariusz Alternatywny 2

- 1–5. Jak w scenariuszu głównym.
- [Błąd synchronizacji z CKAN]
- 6b. System wykrywa błąd podczas synchronizacji z platformą CKAN.
- 7b. System zapisuje zmiany lokalnie i wyświetla komunikat o błędzie synchronizacji.
8. Anulowanie synchronizacji.
- final: *failure*
- POST: Usunięcie zostało zapisane lokalnie, ale nie zsynchronizowane z CKAN.

hasła
stare hasło: tekst
nowe hasło: tekst

4.1.5 Integracja i API



4.1.6 Raportowanie i analiza



UC6006: Lista wygenerowanych raportów

Autor: Ilya Rohoutseu

Scenariusz Główny

PRE: Użytkownik danych jest zalogowany. Wyświetlone jest menu główne.

1. Użytkownik danych wybiera opcję "Lista raportów".
2. System pobiera listę wszystkich wygenerowanych raportów z bazy.

[Pobieranie udane]

[Raporty istnieją]

3. System wyświetla listę raportów z informacjami o dacie, typie oraz innymi metadanymi.

final: *success*

POST: Lista raportów została wyświetlona.

Scenariusz Alternatywny 1

- 1-2. Jak w scenariuszu głównym.

[Błąd pobierania danych]

- 3a. System wyświetla komunikat informujący o błędzie.

final: *failure*

POST: Lista raportów nie została wyświetlona.

Scenariusz Alternatywny 2

1-2. Jak w scenariuszu głównym.

[Brak raportów]

3b. System wyświetla komunikat informujący o braku wygenerowanych raportów..

final: *failure*

POST: Lista raportów nie została wyświetlona.

Słownik

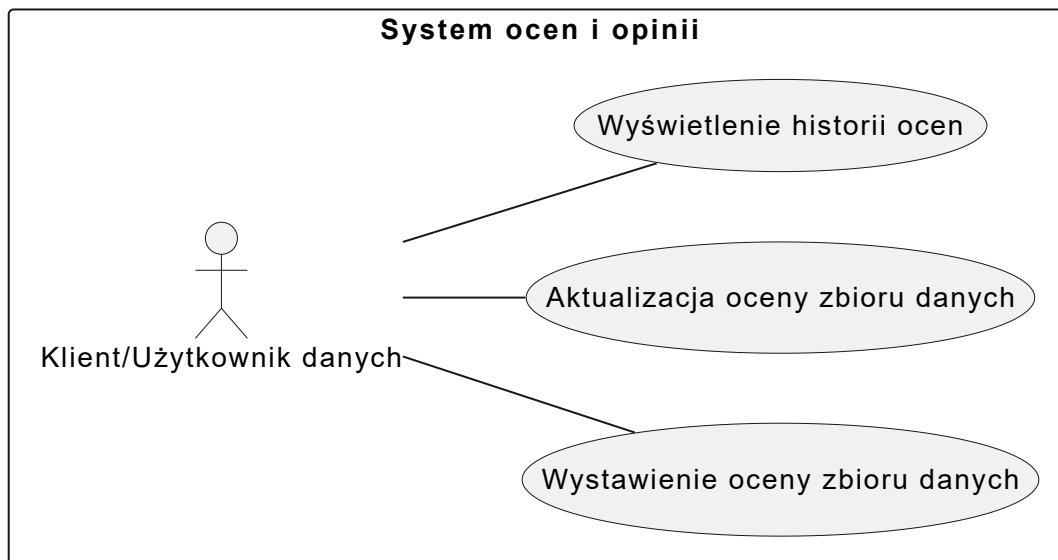
Raport

- **Nazwa:** tekst
- **Data:** date
- **Rozmiar:** liczba
- **Opis:** tekst

Komunikat

- **Nazwa:** tekst
- **Treść:** tekst

4.1.7 System ocen i opinii



UC7001: Wystawienie oceny zbioru danych

Autor: Stanisław Yarmats

SCENARIUSZ GŁÓWNY

PRE: Użytkownik jest zalogowany i wyświetla listę zbiorów danych

1. Użytkownik wybiera opcję "Oceń zbiór"
2. System wyświetla formularz wystawienia oceny
3. Użytkownik wprowadza ocenę i opcjonalnie komentarz
4. System waliduje przesłane dane [dane poprawne]
5. System zapisuje ocenę i aktualizuje średnią ocen zbioru [zapisanie danych powiodło się]
6. System wyświetla komunikat potwierdzający pomyślne wystawienie oceny

final: success

POST: Ocena i/lub komentarz zostały zapisane. Średnia ocen jest zaktualizowana

SCENARIUSZ ALTERNATYWNY 1 – Nieprawidłowa wartość oceny

1.-4. tak jak w SCENARIUSZU GŁÓWNYM
[nieprawidłowa wartość oceny]

5a. System wyświetla komunikat o nieprawidłowym formacie lub zakresie oceny [anulowanie wystawienia oceny]

6a. Użytkownik wybiera opcję "Zamknij"

7a. System zamyka formularz wystawienia oceny

final: cancel

POST: Ocena i/lub komentarz nie są zapisywane. Średnia ocen pozostaje bez zmian

SCENARIUSZ ALTERNATYWNY 2 – Błąd podczas zapisu do bazy danych

1.-5. tak jak w SCENARIUSZU GŁÓWNYM
[błąd podczas zapisu do bazy danych]

6a. System wyświetla komunikat o błędzie bazy danych [anulowanie wystawienia oceny]

7a. Użytkownik wybiera opcję "Zamknij"

8a. System zamyka formularz wystawienia oceny

final: failure

POST: Ocena i/lub komentarz nie są zapisywane. Średnia ocen pozostaje bez zmian

Formularz wystawienia oceny
ocena : int komentarz : string

Komunikat pomyślnego wystawienia oceny
wartość : string

UC7002: Aktualizacja oceny zbioru danych

Autor: Ivan Rusinovich

Opis: System umożliwia użytkownikowi aktualizację wcześniej wystawionej oceny, w tym zmianę wartości oraz komentarza.

SCENARIUSZ GŁÓWNY

PRE: Użytkownik jest zalogowany i listę zbioró danych, istnieje zbiór dla którego wcześniej wystawił ocenę

1. Użytkownik wybiera opcję aktualizacji oceny dla zbioru danych
2. System wyświetla formularz z dotychczasową oceną i komentarzem
3. Użytkownik modyfikuje wartość oceny oraz/lub komentarz w formularzu
4. Użytkownik zatwierdza zmiany
5. System waliduje zmodyfikowane dane (wartość oceny, komentarz)
[dane poprawne]
6. System zapisuje zaktualizowaną ocenę i przelicza średnią ocen zbioru
[zapisanie danych powiodło się]
7. System wyświetla komunikat potwierdzający pomyślną aktualizację oceny

final: success

POST: Ocena i/lub komentarz zostały zaktualizowane. Średnia ocen zbioru jest przeliczona

SCENARIUSZ ALTERNATYWNY 1 – Nieprawidłowa wartość oceny lub komentarz

1.-5. tak jak w SCENARIUSZU GŁÓWNYM

[nieprawidłowa wartość oceny lub komentarz]

6a. System wyświetla komunikat o nieprawidłowym formacie lub zakresie wprowadzonych danych
[anulowanie aktualizacji oceny]

7a. Użytkownik wybiera opcję "Anuluj" lub "Zamknij"

8a. System zamyka formularz aktualizacji oceny

final: cancel

POST: Ocena i/lub komentarz nie są aktualizowane. Średnia ocen zbioru pozostaje bez zmian

SCENARIUSZ ALTERNATYWNY 2 – Błąd podczas zapisu zaktualizowanej oceny

1.-6. tak jak w SCENARIUSZU GŁÓWNYM

[błąd podczas zapisu zaktualizowanej oceny]

7b. System wykrywa błąd przy zapisie zmienionej oceny i wyświetla komunikat o błędzie
[anulowanie aktualizacji oceny]

8b. Użytkownik wybiera opcję "Anuluj" lub "Zamknij"

9b. System zamyka formularz aktualizacji oceny

final: failure

POST: Ocena i/lub komentarz nie są aktualizowane. Średnia ocen zbioru pozostaje bez zmian

Formularz aktualizacji oceny

ocena : int
komentarz : string

Komunikat pomyślnej aktualizacji oceny
--

treść : string

UC7003: Wyświetlenie historii ocen zbioru danych

Autor: Oskar Biwejnias

Opis: System umożliwia użytkownikowi wyświetlenie ocen dotyczących danego zbioru danych

SCENARIUSZ GŁÓWNY

PRE: Użytkownik jest zalogowany i widzi listę zbiorów danych

1. Użytkownik wybiera opcję "wyświetl historię ocen" [żądanie przebiegło pomyślnie]
2. System wyświetla historię ocen danego zbioru danych

final: success

POST: Historia ocen jest wyświetlona

SCENARIUSZ ALTERNATYWNY 1 – Błąd podczas pobierania historii ocen

PRE: Użytkownik jest zalogowany i widzi listę zbiorów danych

1. Jak w scenariuszu głównym [wystąpił błąd] 2a. System wyświetla komunikat o błędzie

final: failure

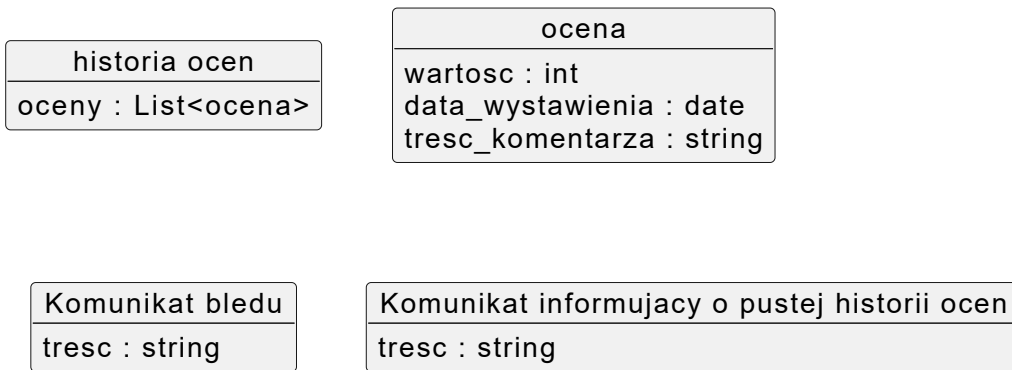
POST: Historia ocen nie jest wyświetlona

SCENARIUSZ ALTERNATYWNY 2 – Pusta historia ocen

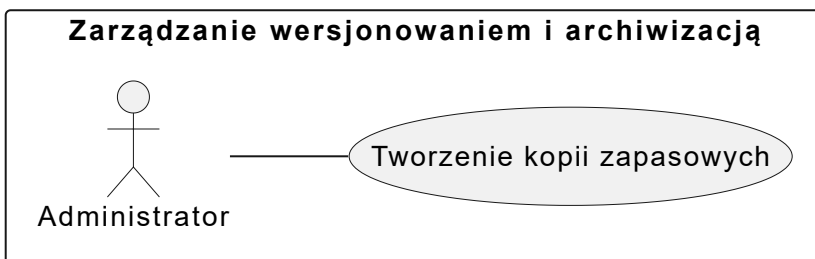
1. Jak w scenariuszu głównym [historia ocen pusta] 2b. System wyświetla komunikat o pustym zbiorze

final: failure

POST: Historia ocen nie jest wyświetlona



4.1.8 Zarządzanie wersjonowaniem i archiwizacją



4.2 Wymagania jakościowe i ograniczenia

Ograczenia

(O0001) Ze względu na ograniczenia budżetowe, stworzona zostanie jedna implementacja interfejsu użytkownika, która będzie adaptowała się do różnych wyświetlaczy.

Wymagania jakościowe

(J0001) Adaptacja interfejsu użytkownika do wyświetlaczy o różnych rozmiarach

Typ	<i>Użyteczność</i>
Priorytet	<i>ważne</i>
Trudność	<i>średnia</i>
Sposób pomiaru	Weryfikacja interfejsu na zestawie różnych urządzeń oraz symulatorów/emulatorów o wybranych rozdzielczościach ekranów zgodnie z przygotowanym scenariuszem testowym.
Oczekiwane wartość	Interfejs automatycznie dopasowuje się do różnych rozmiarów ekranów bez utraty funkcjonalności lub czytelności. Brak istotnych problemów wizualnych i funkcjonalnych na min. 95% testowanych rozdzielczości ekranów urządzeń docelowych.

(J0002) Obsługa trybu ciemnego i adaptacja trybu wyświetlania do ustawień trybu wyświetlania systemu operacyjnego

Typ	<i>Użyteczność</i>
-----	--------------------

Priorytet	<i>przydatne</i>
Trudność	<i>średnia</i>
Sposób pomiaru	Weryfikacja wizualna oraz testy zgodności interfejsu z ustawieniami wyświetlania trybu ciemnego/ jasnego w systemach operacyjnych (Windows, macOS, Android, iOS, Linux).
Oczekiwane wartość	Interfejs automatycznie dostosowuje się do ustawień trybu wyświetlania systemu operacyjnego bez konieczności ręcznej ingerencji użytkownika. Brak błędów wizualnych (np. problemy z czytelnością/kontrastem) dla co najmniej 95% ekranów.

(J0003) Czas obsługi widoków interfejsu użytkownika przez nowych i doświadczonych użytkowników

Typ	<i>Użyteczność</i>
Priorytet	<i>ważne</i>
Trudność	<i>średnia</i>
Sposób pomiaru	Pomiar czasu obsługi ekranów przez grupy użytkowników (nowi, doświadczeni).
Oczekiwane wartość	Średni czas obsługi ekranu dla nowych użytkowników poniżej 2 minut, poniżej 1 minuty dla doświadczonych (> miesiąc używania systemu).

(J0004) Dostępność interfejsu użytkownika dla osób z niepełnosprawnościami

Typ	<i>Użyteczność</i>
Priorytet	<i>ważne</i>
Trudność	<i>średnia</i>
Sposób pomiaru	Audyt zewnętrzny dostępności interfejsu użytkownika zgodnie ze standardami (np. WCAG).
Oczekiwane wartość	System spełnia wymagania dostępności ekranów publicznych, posiada przełącznik umożliwiający włączenie funkcji dostępności.

(J0005) Aktualizacje systemu i jego utrzymanie realizowane w sposób niewywołujący przerw technicznych

Typ	<i>Niezawodność</i>
Priorytet	<i>ważne</i>
Trudność	<i>wysoka</i>

Sposób pomiaru	Przeprowadzenie aktualizacji systemu zgodnie ze scenariuszem testowym. Monitorowanie działania systemu i usług w trakcie aktualizacji i bezpośrednio po niej, przy pomocy narzędzi monitorujących.
Oczekiwane wartość	Proces aktualizacji oraz rutynowe działania utrzymaniowe nie powodują wystąpienia przerw technicznych widocznych dla użytkowników. Dostępność funkcjonalności dla użytkownika końcowego podczas aktualizacji wynosi co najmniej 99,9%.

(J0006) Poprawne działanie systemu przy wyłączeniu z użytku 50% serwerów

Typ	<i>Niezawodność</i>
Priorytet	<i>ważne</i>
Trudność	<i>wysoka</i>
Sposób pomiaru	Testy odporności systemu polegające na kontrolowanym wyłączeniu 50% serwerów zgodnie z wcześniej przygotowanym scenariuszem testowym; monitorowanie dostępności oraz użyteczności systemu.
Oczekiwane wartość	System zachowuje pełną podstawową funkcjonalność i działa stabilnie, bez istotnego pogorszenia wydajności oraz bez błędów krytycznych dla użytkowników końcowych, przy jednoczesnym wyłączeniu do 50% serwerów.

(J0007) Maksymalny czas awarii (MTTR)

Typ	<i>Niezawodność</i>
Priorytet	<i>krytyczne</i>
Trudność	<i>wysoka</i>
Sposób pomiaru	Pomiar czasu usunięcia awarii losowych lub symulowanych awarii
Oczekiwane wartość	Średni maksymalny czas naprawy awarii (MTTR) wynosi 2 godziny

(J0008) Automatyczne mechanizmy wykrywania, przywracania oraz replikacji stanu systemu

Typ	<i>Niezawodność</i>
Priorytet	<i>ważne</i>
Trudność	<i>wysoka</i>
Sposób pomiaru	Regularne audyty, symulacje awarii oraz testy niezawodnościowe środowiska systemowego, obejmujące automatyczne wykrywanie awarii, samoczynne przywracanie oraz automatyczną replikację danych i stanu aplikacji zgodnie z opracowanymi scenariuszami testowymi.

Oczekiwane wartość	System w co najmniej 95% przypadków w sposób automatyczny wykrywa awarię, rozpoczyna proces przywracania działania oraz zapewnia poprawną automatyczną replikację danych, realizując powrót do stanu stabilnego bez ręcznej ingerencji użytkownika lub administratora.
-------------------------------	--

(J0009) Skalowalność horyzontalna

Typ	<i>Wydajność</i>
Priorytet	<i>przydatne</i>
Trudność	<i>wysoka</i>
Sposób pomiaru	Przegląd architektury systemu (audyt DevOps), testy uruchomienia dodatkowych instancji mikrousług w środowisku testowym.
Oczekiwane wartość	System umożliwia łatwe zwiększenie zasobów obliczeniowych przez dodanie kolejnych instancji. Architektura wspiera auto-skalowanie bez przestojów w działaniu.

(J0010) Krótki czas odpowiedzi systemu przy wielu sesjach jednocześnie

Typ	<i>Wydajność</i>
Priorytet	<i>ważne</i>
Trudność	<i>wysoka</i>
Sposób pomiaru	Testy obciążeniowe systemu,
Oczekiwane wartość	System odpowiada użytkownikom w średnim czasie mniejszym niż 1 sekunda przy 100 uruchomionych sesjach jednocześnie

(J0011) Strategia kopii zapasowych danych

Typ	<i>Niezawodność</i>
Priorytet	<i>ważne</i>
Trudność	<i>średnia</i>
Sposób pomiaru	Przegląd logów kopii zapasowych, testy przywracania danych z kopii zapasowych dziennych, tygodniowych i miesięcznych.
Oczekiwane wartość	Kopie zapasowe tworzone codziennie (14 dni), tygodniowo (2 miesiące), oraz miesięczne (po 2 miesiącach). Minimum 99% skuteczność przywracania danych z kopii zapasowych.

(J0011) Walidacja i blokowanie podejrzanych dokumentów

Typ	<i>Bezpieczeństwo</i>
Priorytet	<i>krytyczne</i>
Trudność	<i>średnia</i>
Sposób pomiaru	Wykonanie testów bezpieczeństwa z użyciem próbnych dokumentów zawierających zewnętrzne odnośniki, osadzone skrypty, makra lub inne nietypowe elementy. Obserwacja, czy system skutecznie wykrywa i odrzuca potencjalnie szkodliwe pliki.
Oczekiwane wartość	System automatycznie odrzuca dokumenty zawierające odstępstwa od standardów (np. osadzone odnośniki HTTP, skrypty JS). Blokada jest komunikowana użytkownikowi z informacją o przyczynie. Wskaźnik skuteczności wykrywania podejrzanych plików: $\geq 95\%$.

(J0013) Czas przetworzenia zapytania do 1 minuty

Typ	<i>Wydajność</i>
Priorytet	<i>ważne</i>
Trudność	<i>średnia</i>
Sposób pomiaru	Pomiar czasu przetwarzania żądań API przy pomocy narzędzi monitorujących. Testy przeprowadzane dla różnych typów danych i warunków obciążenia.
Oczekiwane wartość	Średni czas odpowiedzi systemu nie przekracza 60 sekund, niezależnie od formatu danych. Wartość ta jest mierzona w ramach zautomatyzowanego monitoringu.

(J0014) Czas dodania danych przez UI ≤ 5 minut

Typ	<i>Użyteczność</i>
Priorytet	<i>ważne</i>
Trudność	<i>średnia</i>
Sposób pomiaru	Testy użytkowników końcowych polegające na ręcznym dodaniu nowego źródła danych z wykorzystaniem interfejsu użytkownika.
Oczekiwane wartość	Proces dodania źródła danych przez UI zajmuje użytkownikowi maksymalnie 5 minut – obejmuje logowanie, wybór źródła i zatwierdzenie konfiguracji.

(J0015) Dostępność dokumentacji API

Typ	<i>Wydajność</i>
Priorytet	<i>przydatne</i>
Trudność	<i>niska</i>

Sposób pomiaru	Weryfikacja istnienia i kompletności dokumentacji API (np. w formacie OpenAPI / Swagger). Testy użycia przez osoby trzecie.
Oczekiwane wartość	API posiada pełną, aktualną i publicznie dostępną dokumentację opisującą wszystkie dostępne metody, parametry i schematy odpowiedzi. Dokumentacja pozwala na uruchomienie testowego zapytania w <10 minut.

(J0016) Brak ograniczeń dotyczących rozmiaru plików wejściowych

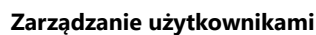
Typ	<i>Wydajność</i>
Priorytet	<i>ważne</i>
Trudność	<i>średnia</i>
Sposób pomiaru	Przeprowadzenie testów funkcjonalnych z użyciem plików o różnych rozmiarach i formatach. Weryfikacja, że system akceptuje i poprawnie przetwarza każdy z nich bez przerw, błędów lub timeoutów.
Oczekiwane wartość	System akceptuje i przetwarza pliki wejściowe dowolnego rozmiaru, bez limitów ilościowych. Brak błędów przetwarzania oraz ograniczeń ze względu na wagę lub strukturę pliku.

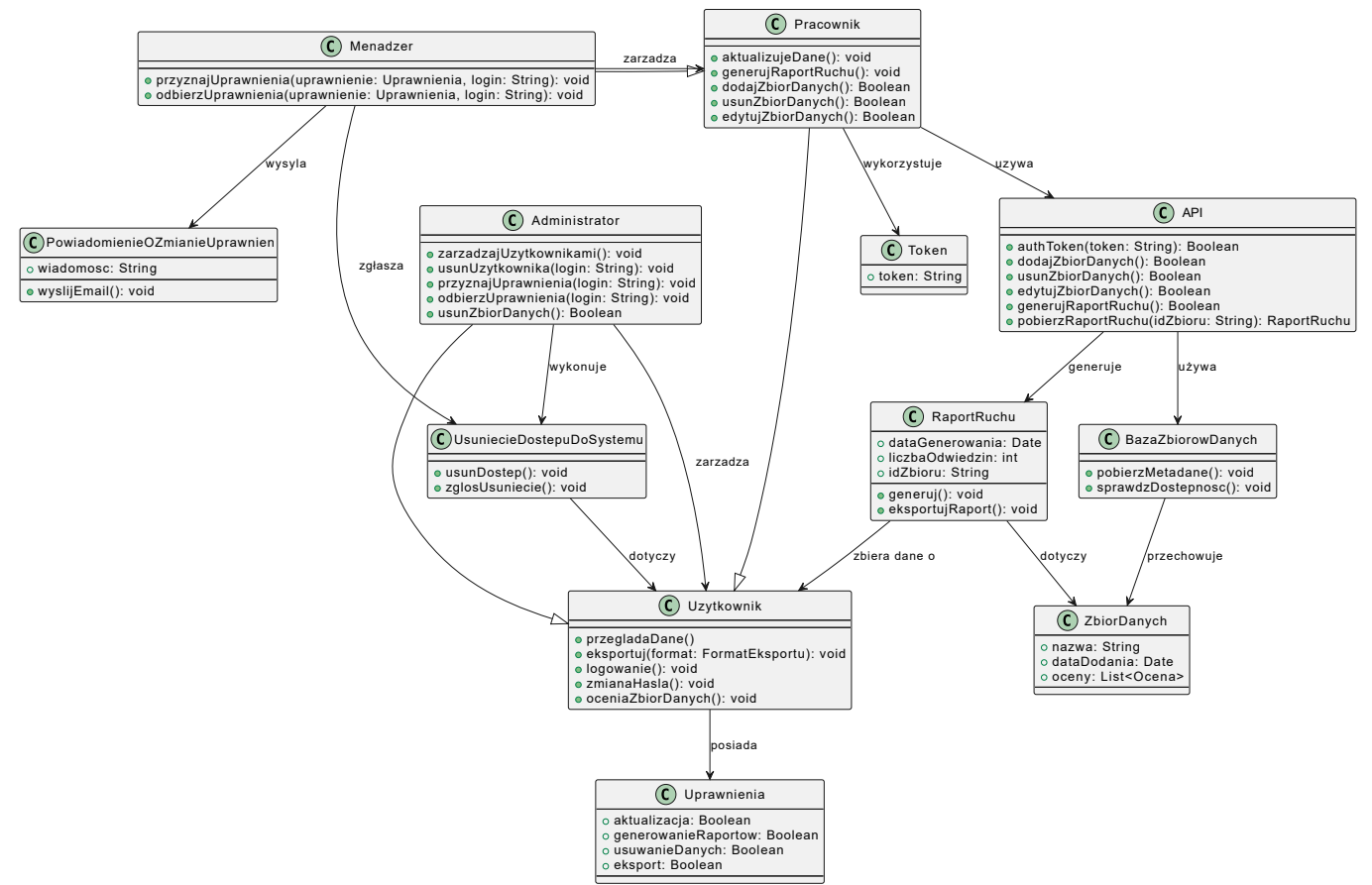
4.3 Słownik dziedziny

Słownik dziedziny

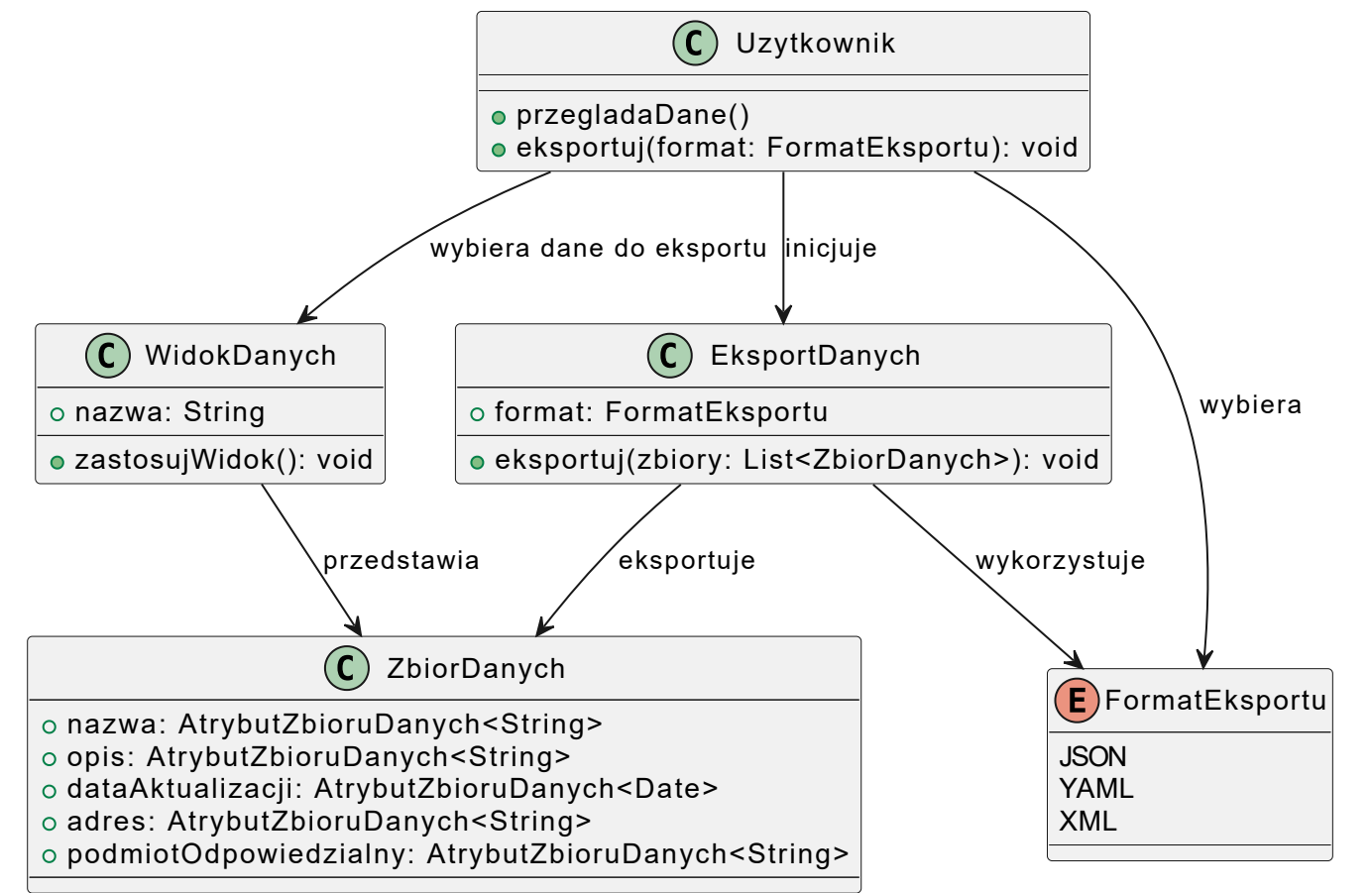
Diagramy

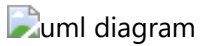
Zarządzanie zbiorami danych



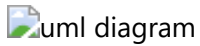


Eksportowanie





Anomalie i archiwizacja



Definicje

Administrator

Użytkownik z najwyższymi uprawnieniami w systemie, odpowiedzialny za zarządzanie **użytkownikami**, dostępami, oraz konfiguracją systemu. Administrator ma pełny wgląd w operacje związane z nadawaniem i odbieraniem uprawnień, dostęp do pełnego audytu systemowego oraz możliwość zarządzania kontami **użytkowników**. Do jego uprawnień należy także **zmiana haseł** użytkowników (w przypadku, gdy ten zapomniał swojego **hasła**), usuwanie dostępu, tworzenie, edytowanie i zarządzanie **schematami danych**.

Adres zbioru

Lokalizacja umożliwiająca dostęp do danego **zbioru danych**. Jest to element widoczny przy **wyświetlaniu zbiorów danych**.

Aktualizacja danych

Proces modyfikowania istniejących danych w systemie. W przypadku aktualizacji przez **API**, zmiany są wprowadzane tylko wtedy, gdy **użytkownik** posiada odpowiednie uprawnienia do dodawania i edytowania danych.

Anomalie

Przypadki, w których dane w systemie odbiegają od ustalonych norm lub założeń, np. nieoczekiwane zmiany w istniejących danych. Może to obejmować błędy, nieprawidłowości lub niezgodności z oczekiwanymi wartościami.

API (Application Programming Interface)

Interfejs umożliwiający komunikację między różnymi systemami, pozwalający na wymianę danych i wykonywanie operacji na danych. API w tym przypadku umożliwia dodawanie i edytowanie zbiorów danych w systemie.

Archiwizacja zbiorów danych

Proces przechowywania kopii zapasowych **metadanych zbiorów danych** w systemie, aby zapewnić możliwość ich odzyskania w przypadku awarii. Archiwizacja obejmuje tworzenie kopii zapasowych zgodnie z ustalonym **harmonogramem**. Same **zbiory danych** nie są objęte archiwizacją.

Archiwizacja danych

Proces, w którym dane są przechowywane w specjalnym, zabezpieczonym miejscu przed ich usunięciem z systemu. Archiwizacja zapewnia, że dane nie zostaną utracone, mimo iż nie są już dostępne w systemie.

Atrybuty zbioru danych (metadane)

Właściwości opisujące [zbiór danych](#), takie jak: [adres zbioru](#), [podmiot odpowiedzialny za zbiór danych](#), [opis, data aktualizacji](#), przechowywane w [bazie zbiorów danych](#). W kontekście [API](#), atrybuty (metadane) są przesyłane i edytowane w formatach takich jak [XML](#) i [JSON](#).

Automatyczne przesyłanie danych

Proces, w którym dane są automatycznie przesyłane do CKAN bez potrzeby ręcznego inicjowania, np. w odpowiedzi na zmiany w systemie.

Baza zbiorów danych

Centralne miejsce przechowywania [metadanych](#) zbiorów danych dostępnych w systemie. System powinien pobierać z bazy danych informacje ([metadane](#)) o wszystkich dostępnych zbiorach danych. W przypadku awarii bazy danych, system powinien informować o jej niedostępności.

Blokada konta

Mechanizm uniemożliwiający użytkownikowi zalogowanie się do systemu, bez usuwania jego danych. Blokada może być nałożona przez [administratora](#) w wyniku decyzji administracyjnej lub naruszenia zasad bezpieczeństwa.

CAPTCHA

System mający na celu weryfikację, czy użytkownik jest człowiekiem, a nie automatycznym botem. Jest częścią procesu [logowanie do systemu](#) i polega na rozwiązaniu prostego zadania (np. zaznaczenie odpowiednich obrazków, odczytaniu tekstu ze zniekształconych liter).

CKAN

Platforma open-source do zarządzania danymi, umożliwiająca ich publikację, integrację z innymi systemami oraz stosowanie standardów tworzenia i udostępniania [schematów danych]. W tym przypadku system synchronizuje swoje dane z CKAN.

Data aktualizacji

Data ostatniej modyfikacji [danego zbioru](#). Jest to element widoczny przy [wyświetlaniu zbiorów danych](#).

Eksport danych z systemu

Proces umożliwiający [użytkownikowi](#) pobranie wybranych danych w formatach [JSON](#) lub [YAML](#). Eksport danych jest dostępny dla wszystkich [użytkowników](#), nie wymagając specjalnych [uprawnień eksportu](#).

Endpoint

Punkt końcowy [API](#), do którego aplikacje mogą wysyłać żądania w celu wykonania operacji na danych (np. dodawanie, edytowanie, pobieranie danych). Każdy endpoint ma określoną funkcję, jaką wykonuje w systemie.

Format danych

Struktura, w jakiej dane są przesyłane do przeglądarki. Format musi być zgodny z wymaganiami przeglądarki oraz odpowiedni dla rodzaju wizualizacji, np. [JSON](#), [XML](#).

Formaty eksportu

Różne struktury, w jakich dane mogą być eksportowane z systemu. Obsługiwane formaty eksportu to [XML](#), [JSON](#), oraz inne formaty, zgodne z wymaganiami użytkowników i systemu.

Gwiazdki

Sposób oceniania, w którym użytkownik wybiera liczbę gwiazdek (od 1 do 5) w celu wyrażenia swojej opinii o danym [zbiorze danych](#). Liczba gwiazdek jest obowiązkowa, a komentarz jest opcjonalny.

Harmonogram archiwizacji

Zasady określające częstotliwość tworzenia kopii zapasowych i czasu ich przechowywania. W systemie obowiązuje następujący harmonogram:

- Kopie codzienne przechowywane przez tydzień.
- Kopie tygodniowe przez kolejne 2 tygodnie.
- Kopie miesięczne przez 6 miesięcy.
- Kopie roczne przechowywane przez 2 lata.

Hasło

Ciąg znaków służący jako sekretne dane do weryfikacji tożsamości użytkownika w procesie logowania. Hasło jest powiązane z [loginem](#) i umożliwia uzyskanie dostępu do systemu. Hasło przy tworzeniu bądź zmianie musi spełniać wymagania dotyczące długości hasła, użycia znaków specjalnych, liczb i wielkich liter.

Historia wyszukiwania zbiorów danych

Zapis wszystkich zapytań wyszukiwania wprowadzonych przez [użytkowników](#). System prowadzi pełny audyt i rejestruje pytania wyszukiwarki. Te statystyki powinny być dostępne dla [administratorów](#).

Integracja z CKAN

Proces umożliwiający wymianę danych pomiędzy systemem zewnętrznym a platformą [CKAN](#), który umożliwia przesyłanie, synchronizowanie i udostępnianie danych w sposób zorganizowany i zgodny z wymaganiami CKAN. Integracja ta może być realizowana w trybie jednokierunkowym (dane przesyłane tylko z systemu do CKAN) lub dwukierunkowym (dane mogą być wymieniane zarówno z systemu do CKAN, jak i odwrotnie). Integracja z CKAN pozwala na automatyczne publikowanie danych w zbiorach danych (datasets), zarządzanie metadanymi, a także na integrację z innymi systemami i aplikacjami, które korzystają z platformy CKAN do

przechowywania i udostępniania danych. Proces ten obejmuje użycie API CKAN, które pozwala na dodawanie, modyfikowanie oraz usuwanie danych, a także na pobieranie informacji o dostępnych zasobach.

JSON

Format wymiany danych, który przechowuje dane w postaci strukturalnej. Jest to format tekstowy oparty na parze klucz-wartość, który jest łatwy do odczytania i zapisania zarówno przez ludzi, jak i maszyny.

Komentarz

Tekstowy opis lub uwaga dodawana przez użytkownika do oceny [zbioru danych](#), który stanowi uzupełnienie dla przypisanej liczby [gwiazdek](#). Komentarz jest opcjonalny, ale jeśli zostanie dodany, nie można go edytować ani usuwać po zapisaniu oceny.

Komunikat o niedostępności bazy danych

Informacja zwracana [użytkownikowi](#) w przypadku awarii [bazy zbiorów danych](#), informująca o braku dostępu do informacji o [zbiorach danych](#).

Kopie zapasowe

Pełne kopie [metadanych zbiorów danych](#), które są tworzone regularnie, w celu ochrony przed utratą danych. Kopie zapasowe są przechowywane zgodnie z ustalonymi zasadami i [harmonogramem](#).

Login

Unikalny identyfikator użytkownika, który jest wykorzystywany podczas logowania do systemu. Login jest wymagany do weryfikacji tożsamości [użytkownika](#) i jest powiązany z odpowiednim [hasłem](#).

Logowanie do systemu

Proces umożliwiający użytkownikowi dostęp do systemu poprzez uwierzytelnienie za pomocą [loginu](#) i [hasła](#). [Użytkownik](#) wprowadza swoje dane, a system weryfikuje poprawność wprowadzonych informacji w celu przyznania dostępu. Logowanie do systemu wymaga rozwiązania zadania [CAPTCHA](#) i może wymagać dodatkowego uwierzytelnienia za pomocą [logowania dwuskładnikowego \(2FA\)](#). System nie obsługuje logowania za pomocą adresu e-mail ani OAuth.

Logowanie dwuskładnikowe (2FA)

Proces uwierzytelniania użytkownika, który wymaga dwóch kroków w celu potwierdzenia tożsamości. Po podaniu [loginu](#) i [hasła](#), [użytkownik](#) musi dodatkowo podać jednorazowy kod przesłany na e-mail.

Komunikat o anomalii

Powiadomienie, które jest wysyłane, gdy wykryta zostaje anomalia w systemie. Komunikat informuje [administratorów](#) o wystąpieniu incydentu, ale nie wskazuje bezpośrednio osoby odpowiedzialnej.

Komunikat o błędzie zmiany hasła

Komunikat wyświetlany użytkownikowi, informujący o błędach w procesie [zmiany hasła](#), np. w przypadku wprowadzenia dwóch różnych [nowych haseł](#) ("hasło nieprawidłowe").

Menadżer (Przedstawiciel firmy)

Użytkownik reprezentujący podmiot odpowiedzialny za [zbiór danych](#) oraz posiadający uprawnienia do przyznawania dostępu do określonego obszaru systemu. Może nadawać i odbierać uprawnienia użytkownikom związanym z danym obszarem oraz przeglądać audyty dotyczące operacji na [zbiorach danych](#).

Monitorowanie problemów

W przypadku wystąpienia problemu wydajnościowych [administrator](#) decyduje, czy wymaga on monitorowania. Jeśli tak, określa, które wskaźniki wydajnościowe powinny być śledzone.

Nowe hasło

[Hasło](#), które użytkownik wprowadza dwukrotnie, aby je ustalić jako nowe. W przypadku niezgodności obu wprowadzonych nowych haseł, system wyświetli komunikat o błędzie: "hasło nieprawidłowe".

Ocena

Ocena przypisywane do [zbiorów danych](#) przez użytkowników. Ocena przyjmuje formę [gwiazdek](#) z opcjonalnym [komentarzem](#). Ilości [gwiazdek](#) ani [komentarza](#) nie można zmienić. Nie ma możliwości wystawienia opinii dla danej oceny.

Odwiedziny zbioru

Zdarzenie, w którym użytkownik wchodzi na stronę danego zbioru danych w celu jego [przeglądania](#). Dane o odwiedzinach są zbierane anonimowo, bez ujawniania tożsamości odwiedzających.

Operacje na metadanych

Działania wykonywane na metadanych zbiorów danych, takie jak dodawanie, edytowanie lub usuwanie metadanych. Te operacje obsługują formaty takie jak [XML](#) i [JSON](#), które pozwalają na przechowywanie i przesyłanie informacji o zbiorach danych w odpowiednim formacie.

Opis

Krótki tekst charakteryzujący zawartość i przeznaczenie danego [zbioru danych](#). Jest to element widoczny przy [wyświetlaniu zbiorów danych](#).

Podmiot odpowiedzialny za zbiór danych (dostawca danych)

Osoba bądź instytucja dostarczająca dane i odpowiedzialna za tworzenie danego [zbioru danych](#), jego aktualność i poprawność. Jest to element widoczny przy [wyświetlaniu zbiorów danych](#).

Powiadomienie o zmianie uprawnień

Automatyczna wiadomość wysyłana na e-mail'a do [użytkownika](#) informująca o przyznaniu, odebraniu lub zmianie jego uprawnień w systemie.

Proces archiwizacji

Procedura obejmująca tworzenie, przechowywanie i zarządzanie kopiami zapasowymi [metadanych zbiorów danych](#) w systemie. Celem archiwizacji jest zapewnienie bezpieczeństwa danych przez długoterminowe przechowywanie kopii zapasowych.

Pracownik

Użytkownik z podstawowym dostępem do systemu, ograniczonym do funkcji wymaganych do wykonywania obowiązków w danym obszarze. Nie posiada uprawnień do zarządzania innymi użytkownikami.

Przyznanie dostępu do systemu

Proces nadawania użytkownikowi odpowiednich uprawnień w systemie w zależności od jego roli. Przyznawanie dostępu może być wykonywane przez [administratora](#) lub [menadżera](#) uprawnionego do zarządzania dostępem do zbiorów danych dostarczanych przez firmę, dla której pracuje.

Raporty o ruchu

Zestawienia przedstawiające dane o odwiedzinach i interakcjach użytkowników z systemem, takie jak liczba odwiedzin zbiorów danych, w tym dane anonimowe.

Raport wydajności

Zestawienie i analiza dotyczące wydajności systemu, obejmujące dane o działaniu serwerów, baz danych, endpointów, backendu itp. Raporty wydajnościowe nie obejmują monitorowania czasu wykonywania requestów, a zamiast tego koncentrują się na innych aspektach wydajności systemu, takich jak obciążenie serwerów, liczba obsługiwanych zapytań czy stan zasobów systemowych.

Schemat danych

Struktura opisująca sposób organizacji zbioru danych. Może być definiowana w postaci kodu źródłowego, [XML](#), [JSON](#) lub innego formatu zgodnego z [CKAN](#) oraz podlega wersjonowaniu. Podlega śledzeniu zmian w nim wprowadzanych, zachowanie historii modyfikacji oraz umożliwia przywracanie wcześniejszych wersji.

Sortowanie ocen

Możliwość sortowania ocen na podstawie różnych kryteriów, takich jak data wystawienia oceny lub ilość [gwiazdek](#).

Sortowanie wyników wyszukiwania

Uporządkowanie [zbiorów danych](#) zwróconych jako [wyniki wyszukiwania](#), zgodnie z zasadami [sortowania zbiorów danych](#), według wartości w wybranej kolumnie jednego z ich [atrybutów](#).

Sortowanie zbiorów danych

Uporządkowanie [zbiorów danych](#), według kolumny jednego z wyświetlanych [atrybutów zbiorów](#): [adres zbioru](#), [podmiot odpowiedzialny za zbiór danych](#), [opis](#), [data aktualizacji](#).

Statystyki i metryki

Dane liczbowe i mierniki, które pozwalają ocenić wydajność systemu, takie jak obciążenie serwera, czas odpowiedzi endpointów, przepustowość itp.

Stare hasło

[Hasło](#) aktualnie przypisane do konta [użytkownika](#), które jest wymagane do zatwierdzenia [zmiany hasła](#). Użytkownik musi poprawnie wprowadzić stare hasło przed podaniem [nowego hasła](#).

Token

Unikalny identyfikator używany do autoryzacji dostępu do API. Token jest generowany i przekazywany użytkownikowi po weryfikacji jego uprawnień, umożliwiając mu autoryzację i wykonanie operacji na zbiorach danych.

Trafność zapytania

Stopień zgodności [wyników wyszukiwania zbiorów danych](#) z [zapytaniem](#) użytkownika, oceniany na podstawie dopasowania treści [opisów](#) oraz innych [atrybutów zbiorów danych](#) do zapytania.

Typy wizualizacji

Różne formy prezentacji danych, takie jak wykresy, tabele, mapy, diagramy czy inne graficzne reprezentacje danych, które są używane w zależności od charakterystyki [zbioru danych](#) i wymagań interesariuszy.

Uprawnienia do aktualizacji

Zestaw praw dostępu przypisanych [pracownikowi](#), które umożliwiają wykonywanie operacji aktualizacji danych. Aktualizacja nie będzie mogła zostać przeprowadzona bez odpowiednich uprawnień.

Uprawnienia do generowania raportów

Osoby posiadające wyższe uprawnienia, takie jak administratorzy czy uprawnieni pracownicy, mogą generować raporty o ruchu w systemie.

Uprawnienia do usunięcia danych

Zestaw praw dostępu przyznanych użytkownikowi lub administratorowi, umożliwiający usuwanie danych. Tylko administratorzy i pracownicy z odpowiednimi uprawnieniami mogą przeprowadzać operację usuwania danych.

Uprawnienia eksportu

Brak ograniczeń dotyczących [eksportu danych](#). Każdy [użytkownik](#) ma możliwość pobrania danych w formacie [JSON](#) lub [YAML](#) bez specjalnych wymagań dotyczących poziomu uprawnień.

Usunięcie danych

Proces trwałego usuwania danych z systemu. W tym przypadku dane są usuwane, ale przed fizycznym usunięciem są archiwizowane, co pozwala na ich odzyskanie w razie potrzeby.

Usunięcie dostępu do systemu

Proces polegający na odebraniu pewnego poziomu dostępu do systemu, wykonywany wyłącznie przez [administratora](#). Dane użytkownika pozostają w bazie, ale nie ma on dostępu do żadnej funkcjonalności systemu.

Usunięcie użytkownika (dezaktywacja konta)

Proces polegający na trwałym odebraniu użytkownikowi dostępu do systemu bez usuwania jego konta z bazy. Dezaktywowany użytkownik nie może się logować ani korzystać z funkcjonalności systemu, jednak jego dane oraz historia operacji pozostają zachowane w celach audytowych.

Użytkownik

Osoba korzystająca z systemu w celu wyszukiwania i przeglądania [zbiorów danych](#), mająca możliwość eksportu danych w formacie [JSON](#) lub [YAML](#).

Weryfikacja treści

Proces sprawdzania danych pod kątem zgodności z określonymi standardami. W tym przypadku nie jest przeprowadzane sprawdzanie treści danych pod kątem wulgaryzmów.

Widoki danych

Zestaw wstępnie zdefiniowanych sposobów prezentacji danych, które są zaprojektowane przez programistów na podstawie wymagań interesariuszy. Użytkownik nie ma możliwości edytowania lub tworzenia własnych widoków; widoki są ustalane i wdrażane przez zespół deweloperski.

Wizualizacje raportu (szczegóły raportu)

Wykresy, tabele lub inne formy wizualizacji, które przedstawiają wyniki analizy wydajności systemu. Wizualizacje mogą być generowane w przeglądarkach na podstawie zebranych danych.

Wizualizacja zbioru danych (przeglądanie zbioru danych)

Proces przedstawiania [odwiedzanego zbioru](#) w formie graficznej, umożliwiający użytkownikowi łatwiejsze zrozumienie i analizowanie informacji. Wizualizacja może przybierać różne formy, takie jak wykresy, tabele, mapy czy inne przedstawienia danych. Wizualizacje są generowane w przeglądarkach, które oczekują strumienia danych w odpowiednim [formacie danych](#).

Wycofywanie zmian

Wszelkie operacje na [zbiorach danych](#) są rejestrowane, a zmiany nie są usuwane ani cofane. System zapewnia pełny audyt operacji na [metadanych](#), podobnie jak w systemie kontroli wersji (takich jak Git).

Wykrywanie anomalii

Proces monitorowania danych w celu identyfikacji sytuacji, które nie spełniają ustalonych norm. W tym przypadku dane są okresowo walidowane pod kątem zmian, a w przypadku wykrycia anomalii [komunikat](#) o tym zostaje wysłany do [administratorów](#).

Wyniki wyszukiwania zbiorów danych

Podzbiór [zbiorów danych](#), który jest zgodny z wybranymi przez użytkownika [atributami zbiorów danych](#) i jest prezentowany w ramach [wyświetlania zbiorów danych](#). Lista wyników zależy od [trafności zapytania](#), a kolejność wyników może być modyfikowana przez [sortowanie wyników wyszukiwania](#).

Wyświetlanie zbiorów danych

Prezentacja [użytkownikowi](#) tabeli zawierającej [zbiory danych](#) wraz z ich [atributami](#). Lista wyświetlanych [zbiorów danych](#) może obejmować zarówno wszystkie dostępne [zbiory danych](#), jak i [wyniki wyszukiwania zbiorów danych](#). O kolejności i zakresie wyświetlanych zbiorów decydują [trafność zapytania](#) oraz [sortowanie wyników wyszukiwania](#).

YAML

Format danych oparty na strukturze tekstowej, wykorzystywany do przechowywania danych w formie zrozumiałej dla człowieka. Jego składnia jest bardziej przyjazna dla [użytkownika](#) w porównaniu do [JSON](#), używając wcięć do wskazywania hierarchii danych.

XML

Język znaczników służący do przechowywania i wymiany danych w formie tekstowej. XML pozwala na tworzenie własnych znaczników, co sprawia, że jest elastyczny i niezależny od platformy. Dokumenty XML są używane do reprezentowania danych w sposób hierarchiczny, umożliwiając przechowywanie zarówno prostych, jak i złożonych struktur danych. Jest powszechnie stosowany w integracji systemów, bazach danych oraz w przesyłaniu danych między aplikacjami.

Zapytanie

Fraza wprowadzona przez [użytkownika](#) do wyszukiwarki, która jest analizowana przez system w celu dopasowania wyników. System wyszukuje [zbiory danych](#), które mają najlepszą [trafność zapytania](#).

Zgłoszenie usunięcia dostępu

Proces inicjowany przez dział kadr, gdy pracownik traci swoje uprawnienia wynikające z zatrudnienia. Kadry przekazują informację o konieczności usunięcia użytkownika, co skutkuje jego dezaktywacją.

Zmiana hasła użytkownika

Proces umożliwiający użytkownikowi zmianę swojego hasła w systemie. Zmiana hasła wymaga podania [starego hasła](#), [nowego hasła](#) dwukrotnie oraz weryfikacji, czy oba wprowadzone hasła są identyczne i spełniają wymagania bezpieczeństwa. System nie umożliwia zmiany hasła bez znajomości aktualnego [hasła](#), chyba że [administrator](#) wykonuje tę operację w przypadku, gdy użytkownik zapomniał [hasła](#).

Zbiór danych

Kolekcja danych udostępnianych przez systemie. Posiada następujące [atrybuty](#): [adres zbioru](#), [podmiot odpowiedzialny za zbiór danych](#), [opis](#), [data aktualizacji](#).