



AGH

REMEMBER

Uniwersalny kategoryzator

Dokument Specyfikacyjny

Tomasz Marek

IEiT Informatyka niestacjonarne

2021/2022

wersja 1.3

Spis Treści

• Opis	3
• Wymagania	4
• Moduły	6
• Struktura	9

1 Opis

1.1 Wstęp

System uniwersalnego kategoryzatora jest systemem nowym, nie bazuje na żadnym istniejącym rozwiązaniu. W celu zapewnienia rozszerzalności, system powinien być zaprojektowany jako zestaw odrębnych modułów, możliwych do separacji.

Powinien także umożliwiać tryby pracy lokalnej oraz sieciowej (komunikacja z serwerem), oraz umożliwiać migrację na inne platformy (takie jak aplikacja desktopowa, mobilna).

1.2 Środowisko

W pierwotym cyklu system powinien zostać zaimplementowany jako aplikacja przeglądarkowa, komunikująca się z serwerem.

Pozwala to na uniezależnienie aplikacji od systemu operacyjnego co rozwiązuje wiele problemów z kompatybilnością.

Technologie webowe pozwalają także na stosunkowo prostą i nieinwazyjną migrację na inne platformy przy zachowaniu wspólnej bazy kodu.

Fakt ten ułatwia utrzymanie oraz zmniejsza ryzyka związane z rozwojem projektu.

1.3 Ograniczenia

Niekomercyjny charakter projektu sprawia, że projekt nie ma jasno zdefiniowanych ograniczeń biznesowych ani technologicznych.

Jedynymi ograniczeniami są:

- Czas implementacji systemu
- Brak możliwości zastosowania płatnych bibliotek komercyjnych

2 Wymagania

2.1 Wymagania funkcjonalne

- Zarządzanie obiektami
- Przeszukiwanie obiektów
- Filtrowanie obiektów (widok)
- Wizualizacja obiektów
- Zarządzanie tagami
- Zarządzanie kontem użytkownika
- Zarządzanie widokami
- Synchronizacja danych
- Synchronizacja obiektów

Terminologia:

- Zarządzanie - grupuje typowe operacje tworzenia, modyfikacji (zwanej także edycją) oraz usuwania.
- Obiekt (*node*) - w tym kontekście oznacza podstawową jednostkę obsługiwaną przez system (jednostka grupowania danych).

2.2 Wymagania niefunkcjonalne

- Zachowanie norm bezpieczeństwa przechowywania danych użytkownika
- Optymalizacja aplikacji zapewniająca dynamiczną pracę z aplikacją
- Prostota użytkowania
- Czysty, czytelny i zrozumiały interfejs

- Mnogość sposobów wykonania akcji pozwalająca użytkownikowi na szybkie osiągnięcie danego rezultatu w różnych kontekstach

3 Moduły

Moduły są formą organizacji wymaganych funkcjonalności systemu i nie reprezentują aktualnej implementacji podziału.

System można podzielić na 3 główne części.

- Warstwa kliencka - odpowiedzialna za prezentację i interakcję z użytkownikiem.
- Warstwa serwerowa - odpowiedzialna za główną logikę aplikacji
- Warstwa persystencji danych - odpowiedzialna za persystencję danych

3.1 Warstwa kliencka

- Interfejs użytkownika
Prezentacja graficzna aplikacji oraz propagacja akcji klienta do pozostałych modułów
- Moduł komunikacji
Komunikację z częścią serwerową
- Moduł kontroli danych
Lokalny zapis i odczyt danych (w przypadku użycia opcji zapisu lokalnego)
- Moduł synchronizacji danych
Synchronizacja danych w przypadku zmiany zapisu z lokalnego na serwerowy
- Moduł reprezentacji danych
Kalkulacja pozycji obiektów oraz zarządzanie stanem wyświetlania

3.2 Warstwa serwerowa

- Moduł autoryzacyjny/autentykacyjny
Uwierzytelnianie użytkownika, weryfikacja dostępu

- Moduł synchronizacji

Synchronizacja danych pomiędzy urządzeniami

- Moduł komunikacji

Brama pośrednicząca w komunikacji pomiędzy warstwą kliencką i serwerową.

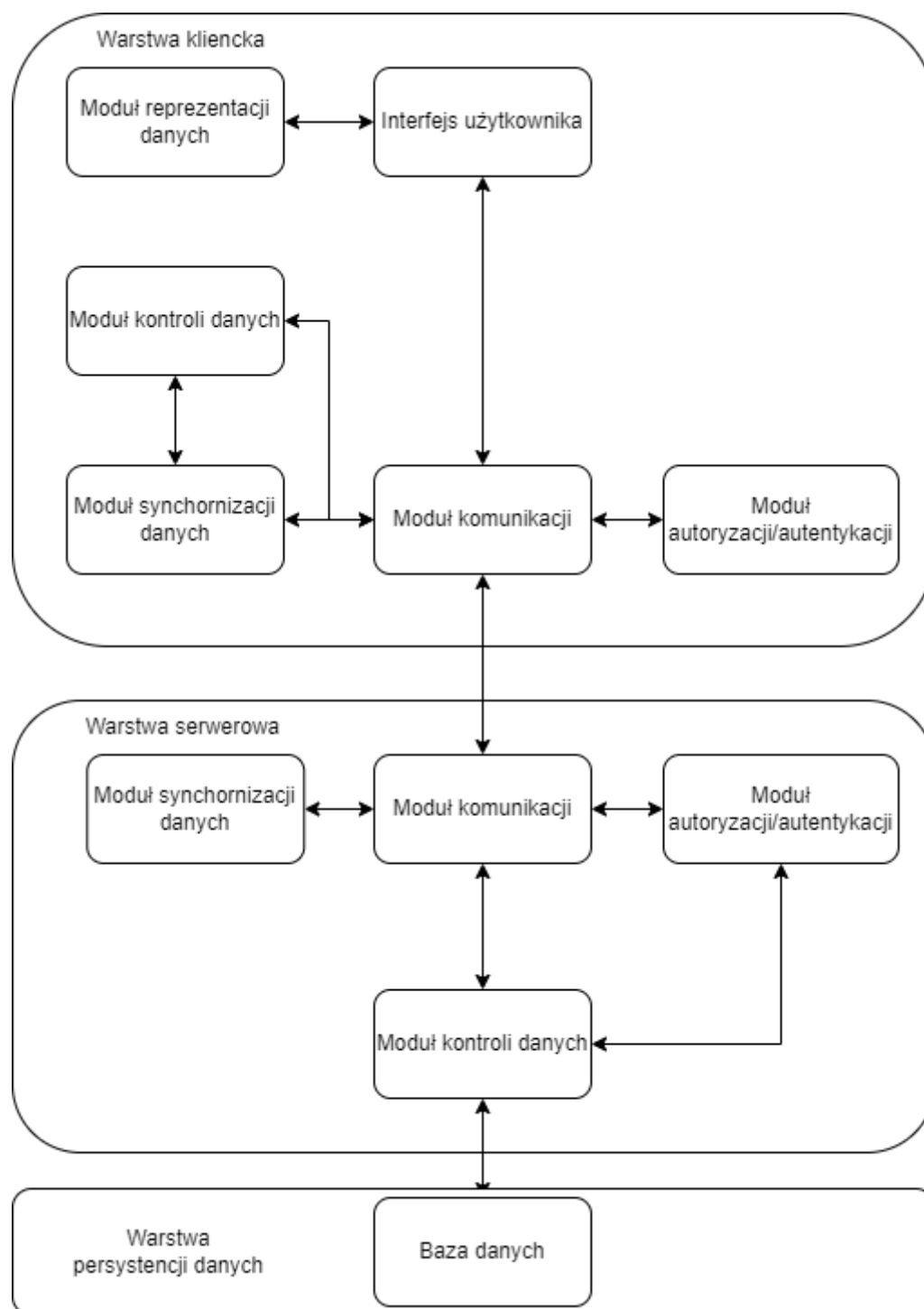
- Moduł kontroli danych

Kontrola zapisu i odczytu danych

3.3. Warstwa persystencji danych

- Baza danych (9)

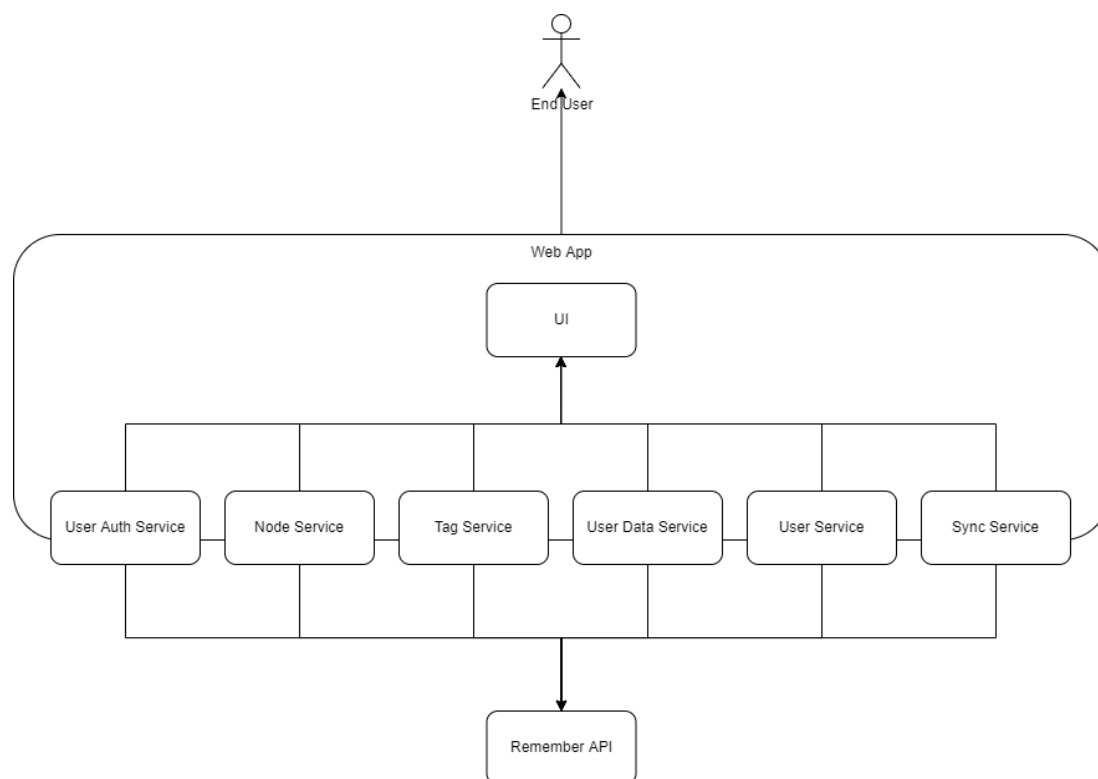
Przechowywanie danych systemu



4 Struktura

4.1 Warstwa kliencka:

4.1.1 Schemat Struktury



4.1.2 Moduły

User Auth

Odpowiedzialny za komunikację z API oraz przechowywanie i zarządzanie danymi autentykacyjnymi użytkownika, zarządzanie sesją.

Node

Odpowiedzialny za komunikację z API, zarządzanie obiektami oraz przechowywanie aktualnego stanu wyświetlania aplikacji.

Tag

Odpowiedzialny za komunikację z API, zarządzania tagami.

User Data

Odpowiedzialny za komunikację z API, oraz przechowywanie danych użytkownika powiązanych z danym modułem.

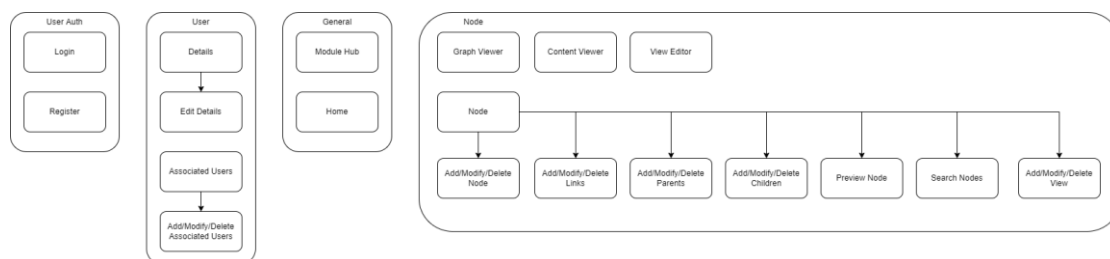
User

Odpowiedzialny za komunikację z API, zwracanie informacji o innych użytkownikach oraz dodawanie powiązań pomiędzy użytkownikami.

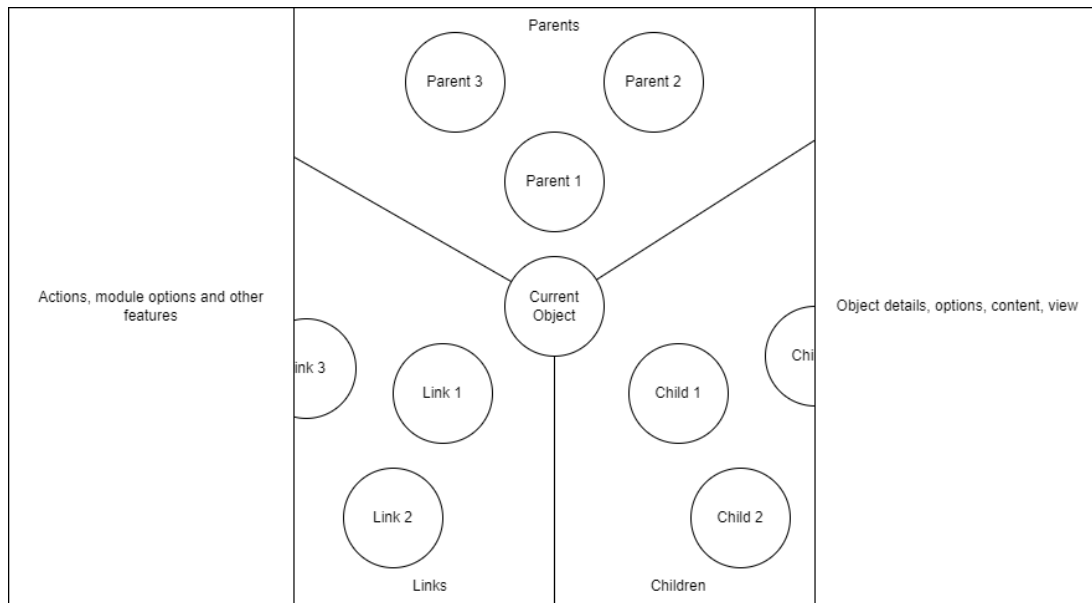
Sync

Odpowiedzialny za dynamiczną synchronizację danych w przypadku ich zmiany niezwiązanej z akcją danego klienta po stronie serwera.

4.1.3 Widoki

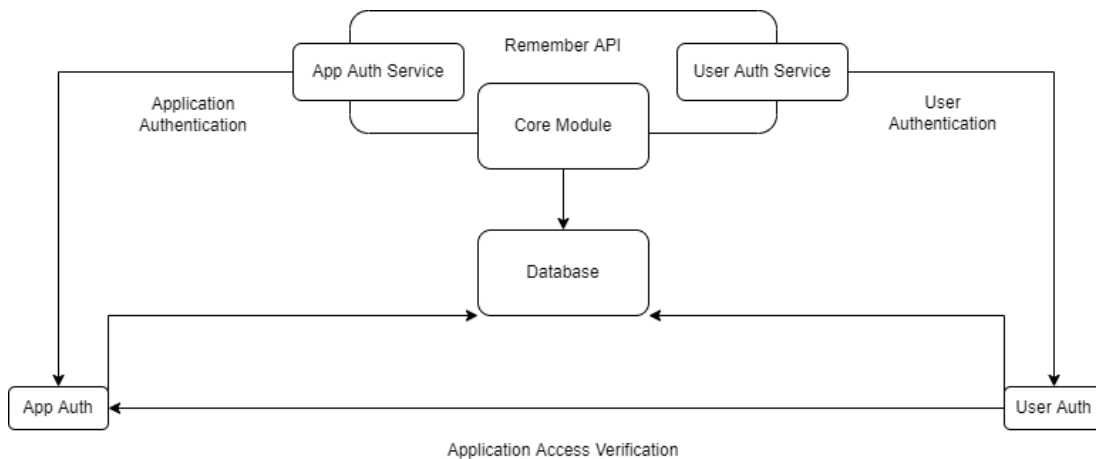


Koncepcja głównego widoku



4.2 Warstwa serwerowa:

4.2.1 Schemat Relacji



4.2.2 Opis Relacji

API powinno posiadać dwa moduły komunikacyjne, kolejno App Auth Service komunikujący się z mikroserwisem App Auth oraz User Auth Service komunikujący się z mikroserwisem User Auth.

Dostęp do mikroserwisu User Auth powinien być możliwy tylko dla aplikacji zarejestrowanych w App Auth, oraz powinien być weryfikowany przy każdorazowym wywołaniu akcji.

Pozwala to na ugenerycznienie serwisów, współdzielenie ich pomiędzy różnymi aplikacjami, systemami, asocjację encji z poszczególnymi aplikacjami, systemami oraz rozszerzalność systemu o nowe serwisy.

API powinno zostać zarejestrowane w App Auth, następnie przy uruchomieniu powinno autentykować się w App Auth. Rezultat powinien umożliwiać autentykację w innych mikroserwisach.

Mikroserwisy po otrzymaniu sygnału wykonania akcji powinny weryfikować tożsamość aplikacji w App Auth.

4.2.3 Opis Modułów

App Auth

Mikroserwis obsługujący kwestie bezpieczeństwa związane z autentykacją aplikacji komunikującej się z mikroserwisami.

Akcje powinny odpowiadać wiadomościom obsługiwanym przez mikroserwis.

Akcje

- Login - akcja służąca do weryfikacji danych logowania aplikacji oraz zwrócenia rezultatu pozwalającego na dalszą identyfikację w systemie.
- Refresh - akcja służąca do odświeżenia "sesji" aplikacji (sesja nie jest tutaj pojęciem dosłownym).
- Verify - akcja służąca do weryfikacji stanu "sesji" aplikacji
- Get - akcja służąca do zwrócenia danych aplikacji

Pojęcie "sesji" nie oznacza dosłownej sesji utrzymywanej pomiędzy klientem i serwerem, oznacza ono przedłużenie sposobu autentykacji (np. przedłużenie ważności tokena lub wygenerowanie nowego).

User Auth

Mikroserwis obsługujący bazowe funkcjonalności związane z autentykacją użytkownika.

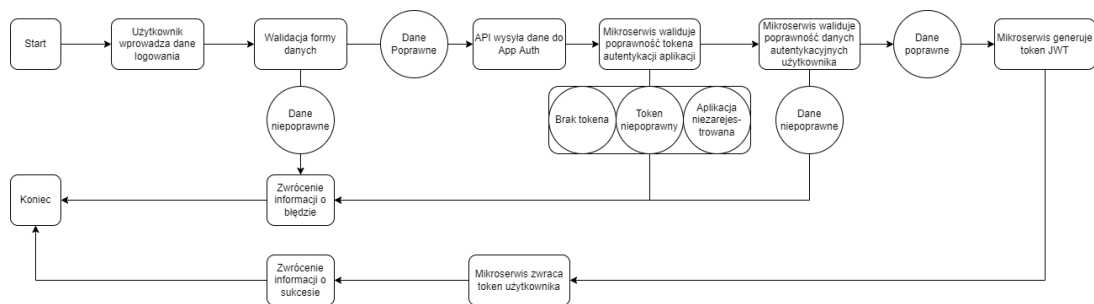
Akcje powinny odpowiadać wiadomościom obsługiwanym przez mikroserwis.

Akcje

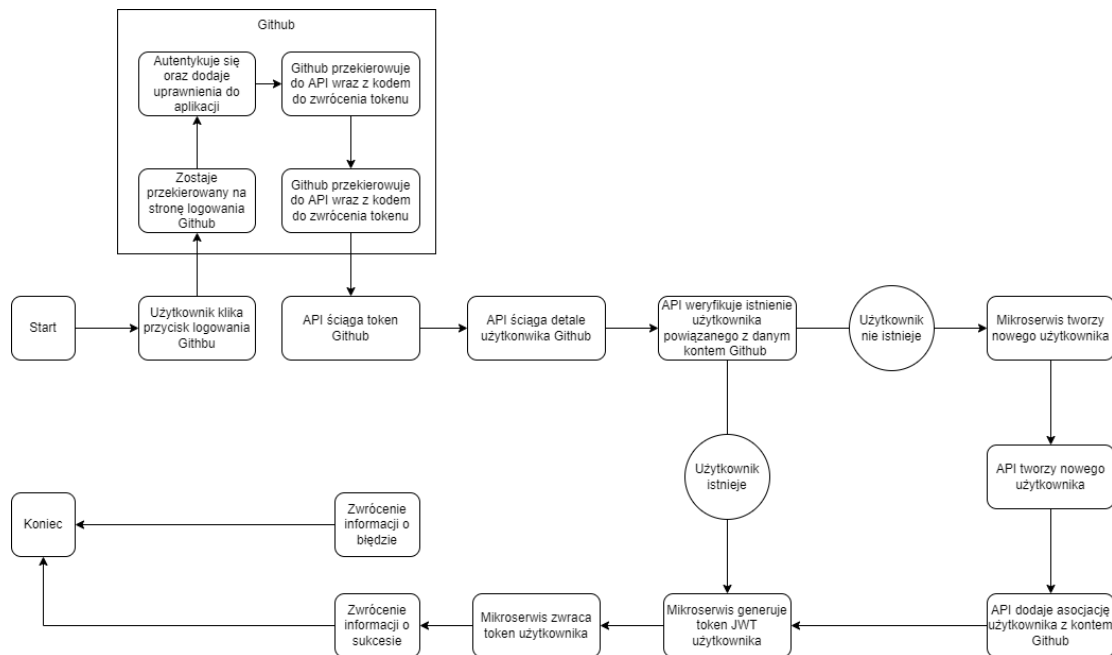
- Login - akcja służąca do weryfikacji danych logowania użytkownika oraz zwrócenia rezultatu pozwalającego na dalszą identyfikację w systemie.
- Verify - akcja służąca do weryfikacji stanu "sesji" użytkownika
- Provider Login - akcja służąca do logowania za pośrednictwem jednego z obsługiwanych dostawców autentykacji/tożsamości.
- Get - akcja służąca do zwrócenia danych użytkownika
- Register - akcja służąca do rejestracji nowego użytkownika

Diagramy

Login



Provider Login (na przykładzie Github)



Remember API

Akcje powinny odpowiadać punktom końcowym uwidacznianym przez API

Podmoduły

App Auth

Podmoduł odpowiedzialny za komunikację z mikroserwisem App Auth.

Przechowuje dane autentykacyjne oraz jest odpowiedzialny za autentykację w innych mikroserwisach.

Akcje

Refresh - akcja służąca do odświeżenia "sesji"

Verify - akcja służąca do weryfikacji stanu "sesji" z App Auth

User Auth

Podmoduł odpowiedzialny za komunikację z mikoserwisem User Auth.

Działa jako wrapper User Auth, rozszerzając bazowe dane oraz akcje w nim zdefiniowane o funkcjonalności wymagane w aplikacji.

Akcje

Login - akcja służąca do logowania użytkownika w systemie

Register - akcja służąca do rejestracji użytkownika w systemie

Refresh - akcja służąca do odświeżenia sesji użytkownika

Get Details - akcja służąca do zwrócenia danych użytkownika

Verify - akcja służąca do weryfikacji stanu sesji użytkownika

Core

Podmoduł zapewniający główną logikę aplikacji.

Należy zwrócić szczególną uwagę na rozgraniczenie akcji na akcje wykonywane na obiektach oraz powiązaniach.

Akcje operujące na powiązaniach nie wpływają bezpośrednio na obiekty (usunięcie powiązania nie usuwa obiektu powiązanego).

Akcje

- Create Node - akcja służąca do utworzenia nowego obiektu
- Update Node - akcja służąca do aktualizacji/edycji obiektu
- Delete Node - akcja służąca do usunięcia obiektu
- Filter - akcja służąca do filtrowania obiektów
- Get Nodes - akcja służąca do zwrócenia wielu obiektów
- Get Node - akcja służąca do zwrócenia obiektu
- Get Node Content - akcja służąca do zwrócenia zawartości obiektu
- Get Node View - akcja służąca do zwrócenia widoku obiektu
- Add Node View - akcja służąca do dodania widoku obiektu

- Update Node View - akcja służąca do aktualizacji widoku obiektu
 - Delete Node View - akcja służąca do usunięcia widoku obiektu
 - Delete Node Links - akcja służąca do usunięcia linków obiektu (powiązań)
 - Add Node Links - akcja służąca do dodania linków obiektu (powiązań)
 - Delete Node Parents - akcja służąca do usunięcia rodziców obiektu (powiązań)
 - Add Node Parents - akcja służąca do dodania rodziców obiektu (powiązań)
 - Delete Node Children - akcja służąca do usunięcia dzieci obiektu (powiązań)
 - Add Node Children - akcja służąca do dodania dzieci obiektu (powiązań)
 - Edit Node Children - akcja służąca do edycji dzieci obiektu (powiązań)
 - Add Node Tags - akcja służąca do dodania tagów obiektu (powiązań)
 - Edit Node Tags - akcja służąca do edycji tagów obiektu (powiązań)
 - Delete Node Tags - akcja służąca do usunięcia tagów obiektu (powiązań)
 - Add Node Permissions - akcja służąca do dodania uprawnień obiektu
 - Edit Node Permission - akcja służąca do edycji uprawnień obiektu
 - Delete Node Permissions - akcja służąca do usunięcia uprawnień obiektu
-
- Add Tag - akcja służąca do dodania tagu
 - Edit Tag - akcja służąca do modyfikacji tagu
 - Delete Tag - akcja służąca do usunięcia tagu
 - Filter Tag - akcja służąca do filtrowania tagów
-
- Get User Data - akcja służąca do zwrócenia danych użytkownika dla modułu
-
- Get Associated Users - akcja służąca do zwrócenia użytkowników powiązanych z konkretnym użytkownikiem
 - Add Associated Users - akcja służąca do dodania asocjacji pomiędzy użytkownikami
 - Delete Associated Users - akcja służąca do usunięcia asocjacji pomiędzy użytkownikami

Node Type - obiekt powinien wyróżniać parę typów takich jak:

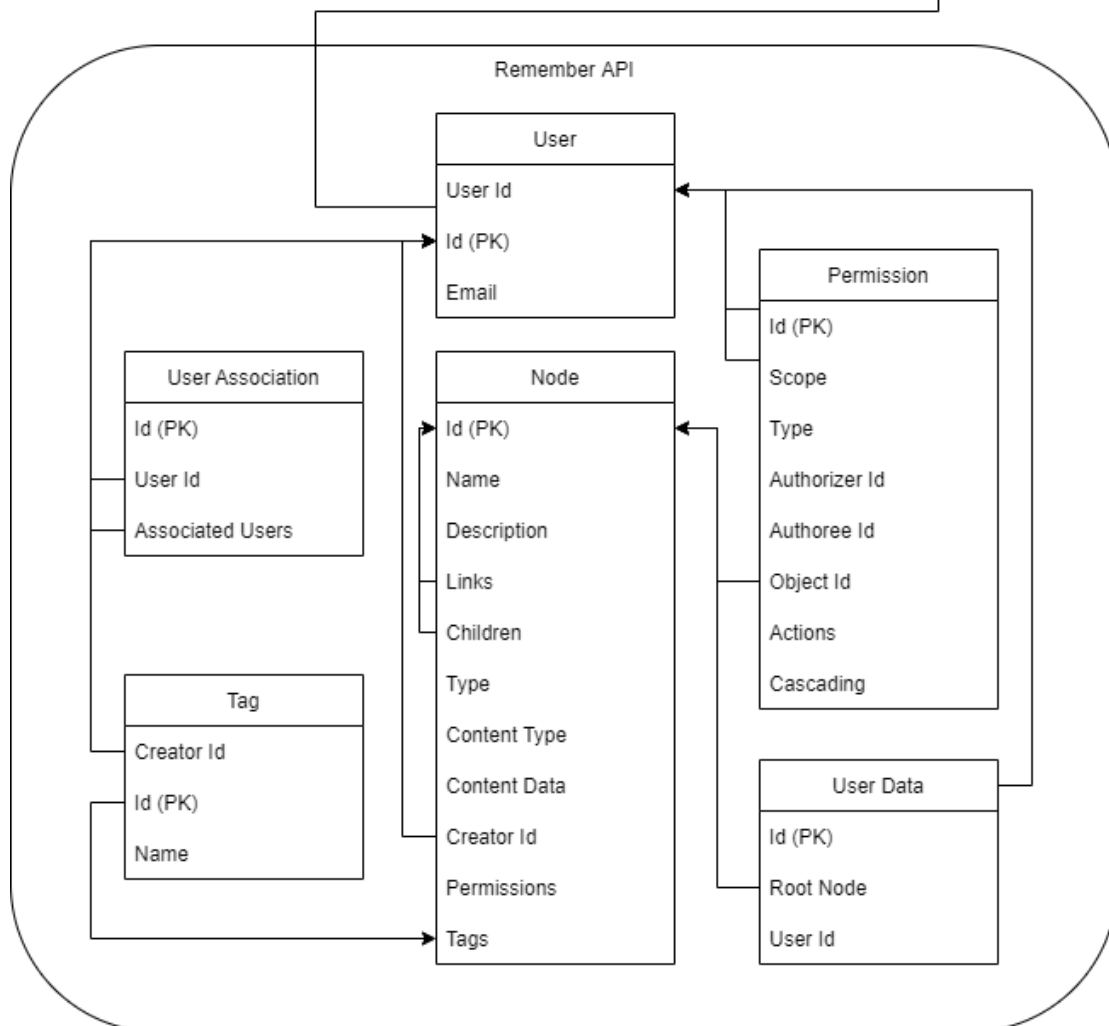
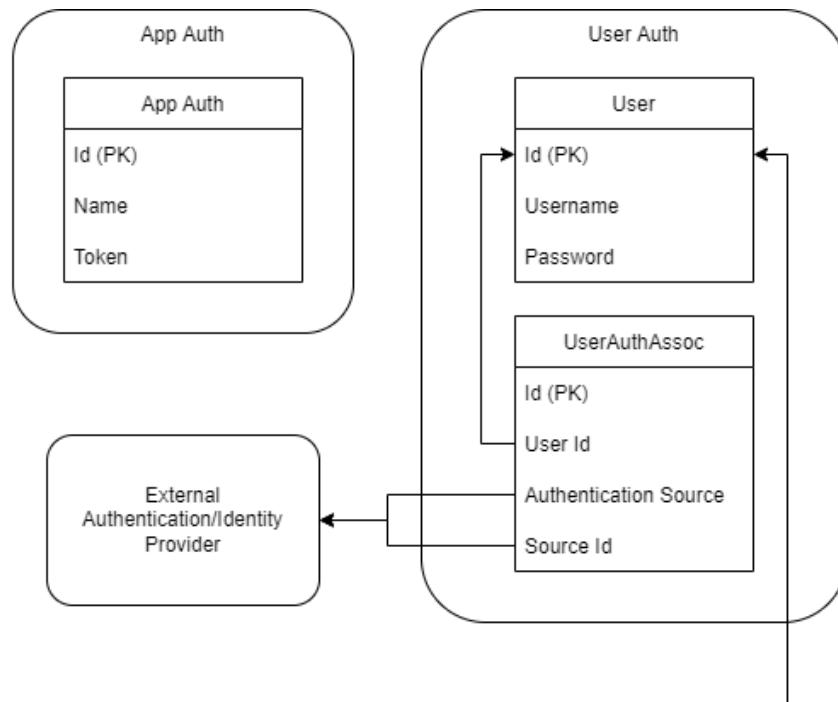
- Root (korzeń) - operacje na tym typie powinny być niedozwolone
- Container - obiekt, który nie zawiera zawartości, służy jako sposób organizacji innych obiektów

- File - obiekt, który zawiera zawartość w postaci pliku
- View - obiekt, który jest widokiem innego obiektu

Node Content Type - określa typ zapisu pliku (czy jest on zapisany jako plik na serwerze czy przechowywany bezpośrednio w bazie danych), oraz jakiego typu jest obiekt Node Content Data.

4.3 Warstwa persystencji danych:

4.3.1 Schemat



4.3.2 Opis schematu

App Auth

App Auth

- Id - Automatycznie generowany identyfikator encji, klucz pierwszy
- Name - Nazwa aplikacji
- Token - Token służący do autentykacji aplikacji

User Auth

User Auth

- Id - Automatycznie generowany identyfikator encji, klucz pierwszy
- Username - Nazwa użytkownika
- Password - Hasło użytkownika

User Auth Association

- Id - Automatycznie generowany identyfikator encji, klucz pierwszy
- User Id - Id użytkownika (User Auth)
- Authentication Source - Wartość pozwalająca na identyfikację zewnętrznego systemu autentykacji/tożsamości.
- Source Id - Identyfikator użytkownika w zewnętrznym systemie autentykacji/tożsamości.

Remember API

User

- Id - Automatycznie generowany identyfikator encji, klucz pierwszy
- User Id - Id użytkownika (User Auth's User)
- Email - Email użytkownika

User Association

- Id - Automatycznie generowany identyfikator encji, klucz pierwszy
- User Id - Id użytkownika (User)
- Email - Email użytkownika

Node (zwane obiektem)

- Id - Automatycznie generowany identyfikator encji, klucz pierwszy
- Name - Nazwa obiektu
- Description - Opis obiektu
- Links - Lista identyfikatorów obiektów powiązanych
- Children - Lista identyfikatorów obiektów będących dziećmi
- Type - Rodzaj obiektu
- Content Type - Rodzaj zawartości obiektu
- Content Data - Dane zawartości obiektu
- Creator Id - Identyfikator twórcy obiektu
- Permissions - Uprawnienia do obiektu
- Tags - Lista identyfikatorów tagów obiektu

Permission

- Id - Automatycznie generowany identyfikator encji, klucz pierwszy
- Scope - Zakres uprawnienia (Publiczny, Prywatny)
- Type - Rodzaj uprawnienia (zezwól, zabroń)
- Authorizer Id - Identyfikator użytkownika nadającego uprawnienie
- Autoree Id - Identyfikator użytkownika, którego to uprawnienie dotyczy
- Object Id - Identyfikator obiektu, którego to uprawnienie dotyczy
- Actions - Akcje, których to uprawnienie dotyczy
- Cascading - Wartość określająca, czy uprawnienie dotyczy także obiektów dzieci

User Data

- Id - Automatycznie generowany identyfikator encji, klucz pierwszy

- Root Node - Identyfikator obiektu będącego korzeniem (głównym obiektem, ang. *root*) użytkownika
- User Id - Identyfikator użytkownika, którego dotyczy ta encja