# Analiza systemu wspomagającego zarządzanie zdjęciami

Grupa nr.4

|  |  |
| --- | --- |
| **Produkt etapu** | **Autor** |
| Przestawienie koncepcji systemu | Emilian Bochenek |
| Specyfikacja funkcjonalna i niefunkcjonala | Daniel Gromak |
| Model danych | Tomasz Jarnutowski |
| Model architektury systemu | Emilian Bochenek |

Spis treści

[Analiza systemu wspomagającego zarządzanie zdjęciami 1](#_Toc121777930)

[Przedstawienie koncepcji systemu 3](#_Toc121777931)

[Opis 3](#_Toc121777932)

[Założenia systemu 3](#_Toc121777933)

[Cele biznesowe 3](#_Toc121777934)

[Plan Złoty 3](#_Toc121777935)

[Plan Srebrny 3](#_Toc121777936)

[Plan Brązowy 4](#_Toc121777937)

[Plan darmowy 4](#_Toc121777938)

[Specyfikacja funkcjonalna i niefunkcjonalna 4](#_Toc121777939)

[Diagram hierarchii funkcji 4](#_Toc121777940)

[Lista funkcji z opisem ich zastosowania 5](#_Toc121777941)

[1. Zarządzanie kontem 5](#_Toc121777942)

[2. Zdjęcia 5](#_Toc121777943)

[3. Foldery 6](#_Toc121777944)

[Opis aktorów 6](#_Toc121777945)

[Diagramy przypadków użycia 7](#_Toc121777946)

[Lista zależności niefunkcjonalnych 8](#_Toc121777947)

[Model danych 9](#_Toc121777948)

[Tabela User 9](#_Toc121777949)

[Tabela Photo 10](#_Toc121777950)

[Tabela Album 10](#_Toc121777951)

[Tabela AlbumPhoto 11](#_Toc121777952)

[Tabela Tag 11](#_Toc121777953)

[Tabela TagPhoto 11](#_Toc121777954)

[Tabela Favourite 12](#_Toc121777955)

[Model architektury systemu 12](#_Toc121777956)

[Diagram 12](#_Toc121777957)

[Opis 13](#_Toc121777958)

[Frontend 13](#_Toc121777959)

[Backend 13](#_Toc121777960)

# 

# Przedstawienie koncepcji systemu

## Opis

Celem systemu jest zarzadzanie biblioteka zdjęć oraz uporządkowanie ich w odpowiednich wyznaczonych dla nich tagach, czy folderach. Ma to za zadanie ułatwić znalezienie oraz zapanowanie nad naszym zbiorem zdjęć.

## Założenia systemu

Głównym założeniem systemu jest przechowywanie zdjęć, dodatkowymi elementami które chcemy wprowadzić aby ułatwić zarządzanie swoją biblioteką zdjęć jest dodawanie tagów do poszczególnych fotografii oraz porządkowanie ich w folderach. Dodatkowymi elementami będzie zbieranie metadanych zdjęcia do automatycznego porządkowania zdjęć wedle lokalizacji, co planujemy zrobić w późniejszych etapach. W pierwszy etapie skupimy się na przechowywaniu zdjęć użytkownika oraz nadawaniu tagów w celu łatwiejszego sortowania.

## Cele biznesowe

Naszym głównym celem biznesowym jest wprowadzenie 3 planów pojemnościowych,

Każdy z planów będzie posiadać większa ilość GB na zdjęcia uzytkownika:

## Plan Złoty

* 1Tb na zdjęcia
* Nielimitowana ilość tagów
* Nielimitowana ilość folderów

## Plan Srebrny

* 500Gb na zdjęcia
* 50 tagów
* 50 folderów

## Plan Brązowy

* 100Gb na zdjęcia
* 25 tagów
* 15 folderów

## Plan darmowy

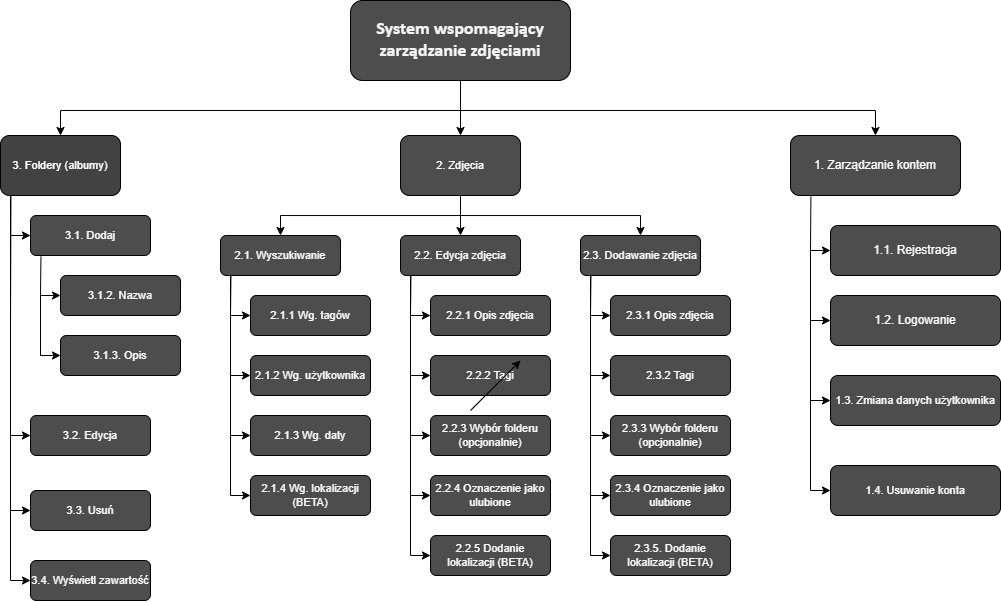
* 15Gb na zdjęcia
* 10 predefiniowanych tagów
* 1 główny folder

Jak widzimy wyżej, każdy z planów ma ustaloną ilość dostępnej przestrzeni na zdjęcia oraz odpowiednią ilość tagów czy folderów.

Znacząco podnosząc ilość przestrzeni oraz dostępności do większej ilości tagów chcemy zachęcić użytkowników do wykupowania subskrypcji. Dodatkowo kupując subskrypcje roczną zaoferujemy 20% zniżki co pozwoli zyskać przyrost środków w początkowej fazie projektu która będziemy mogli przeznaczyć na kolejne udogodnienia.

# Specyfikacja funkcjonalna i niefunkcjonalna

## Diagram hierarchii funkcji



## Lista funkcji z opisem ich zastosowania

### Zarządzanie kontem

#### Rejestracja

Tworzenie użytkownika portalu. Użytkownik, który jeszcze nie ma konta w serwisie może się zarejestrować wypełniając formularz o swoje podstawowe dane t.j. email, login, imie, nazwisko.

#### Logowanie

Proces zalogowania polega na wypełnieniu formularza o nazwę użytkownika i hasło. Po poprawnym uwierzytelnieniu użytkownik logowany jest do portalu.

#### Zmiana danych użytkownika

Proces zmiany danych, które użytkownik podawał o sobie w formularzu rejestracyjnym. Po aktualizacji którejś z danych t.j. e-mail, login, encja użytkownika w systemie zostaje zmieniona o wprowadzone dane.

#### Usuwanie konta.

Usunięcie encji użytkownika z systemu. Usuwa to również wszystkie umieszczone przez niego treści i zdjęcia.

### Zdjęcia

#### Wyszukiwanie

Użytkownik, spośród udostępnionych przez siebie i innych użytkowników zdjęć ma możliwość filtrowania ich po następujących właściwościach zdjęcia:

* Tagi,
* Użytkownik (autor),
* Data,
* Lokalizacja (wykonania) zdjęcia

#### Edycja

Użytkownik może edytować zdjęcia publikowane wyłącznie przez siebie. Parametry zdjęcia, które można zaktualizować to:

* Opis zdjęcia,
* Tagi,
* Folder,
* Oznaczenie jako ulubione,
* Lokalizacja.

#### Dodawanie

Dodawanie treści (zdjęć) jako główna funkcjonalność portalu. Użytkownik może publikować zdjęcia do maksymalnego rozmiaru w ramach posiadanego pakietu. Dodając nowe zdjęcie, użytkownik poza plikiem określa inne dane zdjęcia takie jak:

* Opis zdjęcia,
* Tagi,
* Folder,
* Oznaczenie jako ulubione,
* Lokalizacja.

### Foldery

#### Dodawanie

Dodawanie albumu, w do którego można przypisać publikowane zdjęcia. Podczas tworzenia folderu, użytkownik określa jego następujące dane:

* Nazwa,
* Opis.

#### Edycja

Zmiana następujących parametrów folderu:

* Nazwa,
* Opis.

#### Usuwanie

Usunięcie encji albumu. Podczas tej czynności użytkownik ma możliwość określenia, czy chce również usunąć jego zawartość w postaci zdjęć.

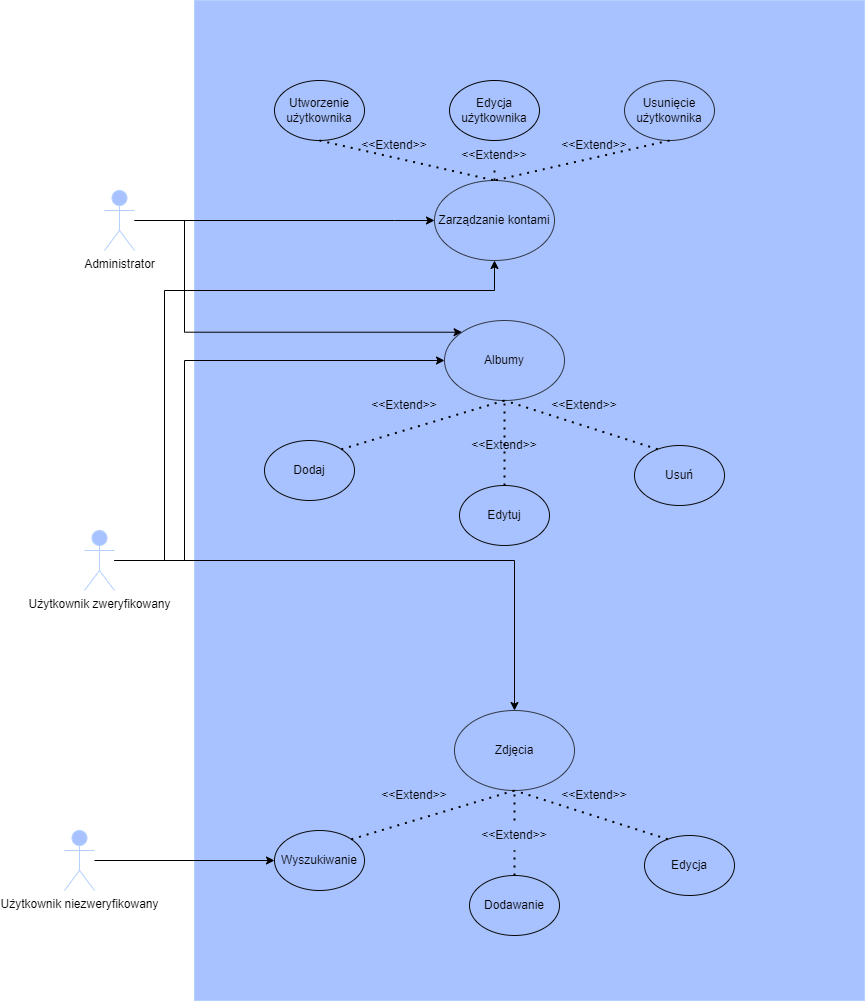
#### Wyświetlanie zawartości

Wyświetlenie zdjęć, które przypisane są do wybranego folderu.

## Opis aktorów

1. Administrator – użytkownik o strukturze encji takiej samej jak zwykły użytkownik, ale z możliwością dodatkowych czynności na portalu dzięki oznaczonej fladze w modelu danych p.t. is\_superuser. Do jego dodatkowych uprawnień należą zarządzanie treściami użytkowników i zarządzanie użytkownikami.
2. Użytkownik zweryfikowany – użytkownik posiadający możliwości przeglądania treści innych użytkowników, a także publikowania swoich zdjęć.
3. Użytkownik niezweryfikowany – użytkownik posiadający możliwość przeglądania treści innych użytkowników.

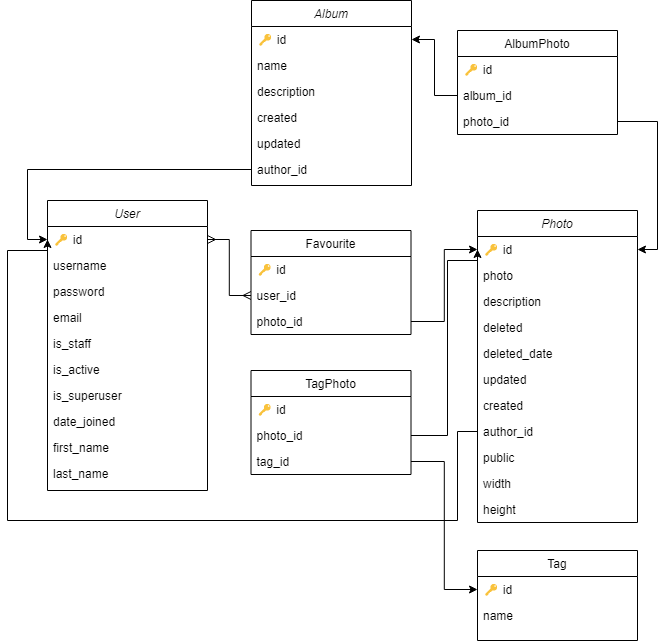
## Diagramy przypadków użycia



## Lista zależności niefunkcjonalnych

1. Kontrola treści – działania zwalczające publikowanie treści uważanych jako nieodpowiednie.
2. Komfort – design zgodny z User Experience, intuicyjny interfejs, kolorami odpowiednio dobrany do pozostałych elementów.
3. Wydajność – gotowość do płynnego działania przy dużym obciążeniu. Portal nie przewiduje potencjalnej liczby użytkowników, lecz jest ona nieograniczona.

# Model danych



## Tabela User

Tabela zawierająca dane o użytkowniku. Jest to domyślna tabela dla użytkownika w frameworku Django

|  |  |  |
| --- | --- | --- |
| **Nazwa** | **Typ** | **Opis** |
| id | Klucz główny, integer | Identyfikator |
| username | varchar, unikalne | Nazwa użytkownika |
| password | varchar | Hasło |
| email | varchar, unikalne | Email |
| is\_staff | bool | Czy jest moderatorem |
| is\_active | bool | Czy konto jest aktywne |
| is\_superuser | boll | Czy jest administratorem |
| date\_joined | datetime | Data utworzenia konta |
| first\_name | varchar | Imię |
| last\_name | varchar | Nazwisko |

## Tabela Photo

Tabela zawierająca informacje o zdjęciu. Posiada klucz obcy author\_id odnoszący się do tabeli User (Relacja typu OneToMany)

|  |  |  |
| --- | --- | --- |
| **Nazwa** | **Typ** | **Opis** |
| id | Klucz główny, integer | Identyfikator |
| photo | varchar | Ścieżka do pliku |
| description | text | Opis |
| public | boolean | Czy zdjęcie jest publiczne |
| deleted | boolean | Czy plik znajduje sie w koszu |
| deleted\_date | datetime | Data umieszczenia pliku w koszu |
| updated | datetime | Data ostatniej aktualizacji |
| created | datetime | Data utworzenia |
| author\_id | Klucz obcy, integer | Id autora zdjęcia |
| width | integer | Wysokość zdjęcia |
| height | integer | Szerokość zdjęcia |

## Tabela Album

Tabela zawierająca informacje o albumie. Posiada klucz obcy author\_id odnoszący się do jedynego autora albumu z tabeli User (Relacja typu OneToMany)

|  |  |  |
| --- | --- | --- |
| **Nazwa** | **Typ** | **Opis** |
| id | Klucz główny, integer | Identyfikator |
| name | varchar | Nazwa albumu |
| description | text | Opis |
| created | datetime | Data utworzenia |
| updated | datetime | Data ostatniej aktualizacji |
| author\_id | Klucz obcy, integer | Id autora albumu |

## Tabela AlbumPhoto

Tabiera zawierająca informację o zdjęciach przypisanych do albumu. Jest to relacja typu ManyToMany ze względu na fakt że każde zdjęcie może mieć wiele albumów

|  |  |  |
| --- | --- | --- |
| **Nazwa** | **Typ** | **Opis** |
| id | Klucz główny, integer | Identyfikator |
| album\_id | Klucz obcy, integer | ID albumu |
| photo\_id | Klucz obcy, integer | ID zdjęcia |

## Tabela Tag

Tabela słownikowa zawierające unikalne nazwy Tagów.

|  |  |  |
| --- | --- | --- |
| **Nazwa** | **Typ** | **Opis** |
| id | Klucz główny, integer | Identyfikator |
| name | varchar, unikalne | Nazwa tagu |

## Tabela TagPhoto

Tabiera zawierająca informację o tagach przypisanych do zdjęć. Jest to relacja typu ManyToMany ze względu na fakt że każde zdjęcie może mieć wiele tagów

|  |  |  |
| --- | --- | --- |
| **Nazwa** | **Typ** | **Opis** |
| id | Klucz główny, integer | Identyfikator |
| tag\_id | Klucz obcy, integer | ID tagu |
| photo\_id | Klucz obcy, integer | ID zdjęcia |

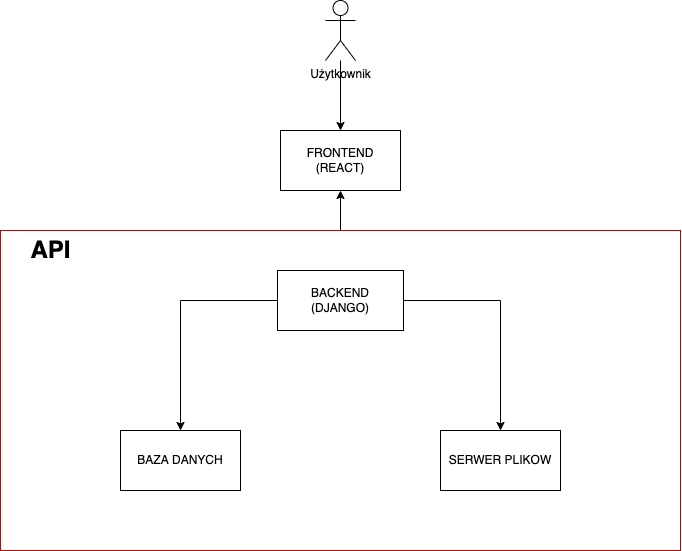
## Tabela Favourite

Tabiera zawierająca informację o zdjęciach dodanych do ulubionych. Jest to relacja typu ManyToMany ze względu na fakt że różni użytkownicy mogą dodać jedne zdjęcie do ulubionych

|  |  |  |
| --- | --- | --- |
| **Nazwa** | **Typ** | **Opis** |
| id | Klucz główny, integer | Identyfikator |
| photo\_id | Klucz obcy, integer | ID zdjęcia |
| user\_id | Klucz obcy, integer | ID użytkownika |

# Model architektury systemu

## Diagram



## Opis

### Frontend

Zaprojektowany będzie w React, będziemy dostarczać użytkownikowi wygenerowaną wcześniej stronę bez potrzeby przeprowadzania obliczeń za pomocą serwera frontendowego, zredukuje to czas załadowania strony i pozwoli nam na utrzymywanie większej ilości użytkowników.

### Backend

Napisany w django, będą tutaj wykonywane wszystkie obliczenia oraz komunikacja z baza danych oraz serwerem plików do przechowywania zdjęć.

Backend będzie tworzyć role API w naszej strukturze.