

Dokumentacja projektu wykonanego w ramach zajęć

## **BAZY DANYCH I**

Alan Guzek

21.01.2023

## **I. Projekt koncepcji, założenia**

### **1. Zdefiniowanie tematu projektu**

Tematem projektu jest internetowa aplikacja bazodanowa pozwalająca tworzyć potyczki dla rozgrywek RPG, a następnie pozwalać na ich przegląd.

### **2. Analiza wymagań użytkownika**

Projektowana aplikacja będzie zezwalała Administratorowi na dodawanie elementów potyczek takich jak potwory, czy skarby oraz ich usuwanie.

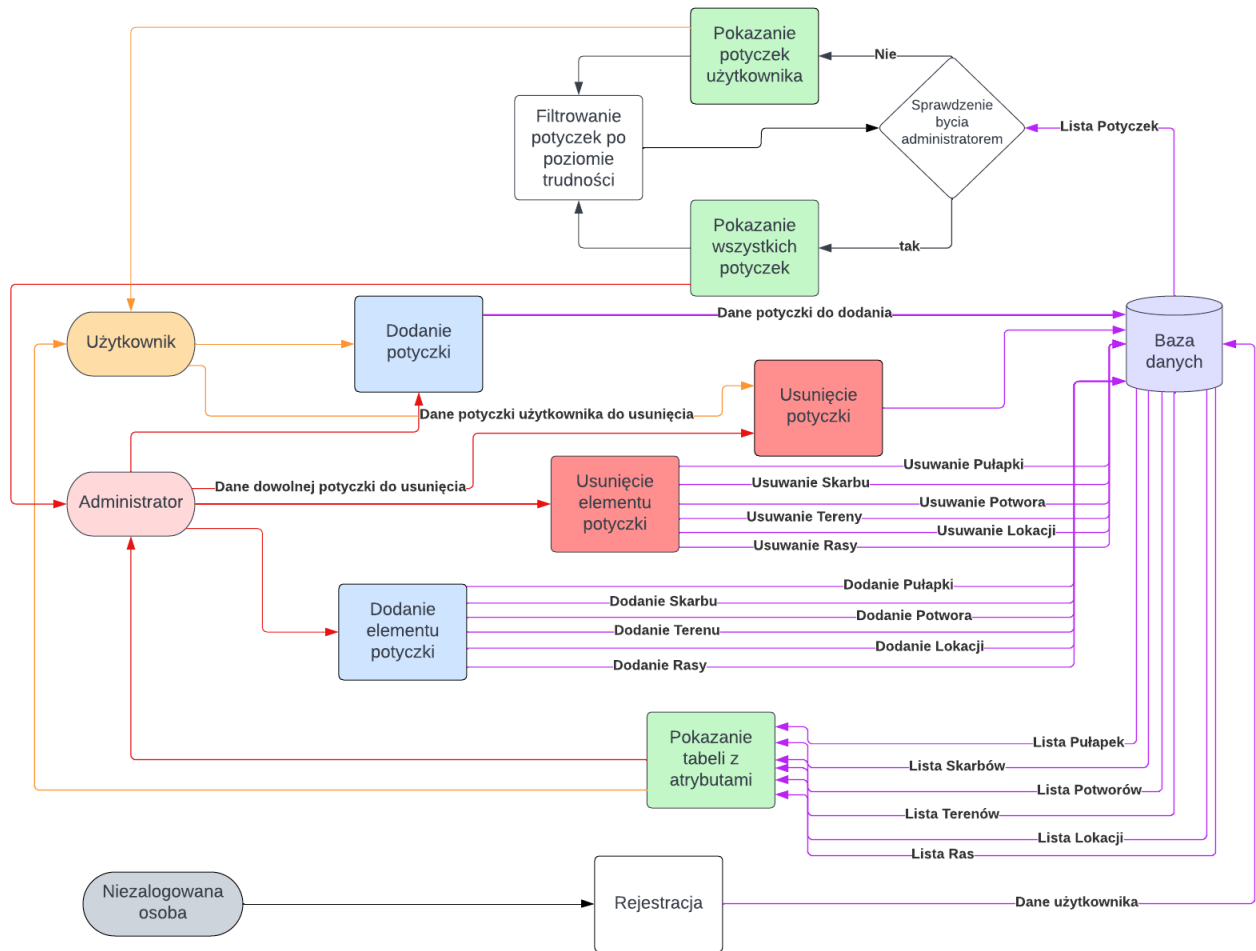
Użytkownik będzie mógł jedynie zarządzać swoimi potyczkami.

### **3. Zaprojektowanie funkcji**

Podstawowymi funkcjami będzie dodawanie potyczek składających się z różnych elementów oraz ich przeglądanie i usuwanie. Dodatkowo będzie można się w serwisie zarejestrować oraz zalogować, natomiast Administrator będzie miał dostęp do większej liczby tabel.

## **II. Projekt diagramów**

### **4. Budowa i analiza diagramu przepływu danych (DFD)**



## 5. Zdefiniowanie encji oraz ich atrybutów

- \* tereny – możliwe tereny do zamieszkiwania. Zawierają nazwę oraz opis.
- \* lokacje – miejsca, które znajdują się na określonych terenach. Zawierają nazwę oraz opis.
- \* rasy – różne rasy potworów, które mają określone tereny, na których żyją. Zawierają nazwę oraz opis.
- \* potwory – przeciwnicy w potyczkach, którzy mają odpowiednią rasę. Zawierają nazwę, opis oraz poziom trudności.
- \* skarby – możliwe nagrody za wygranie potyczki. Zawierają nazwę, opis, rzadkość oraz wartość.

\* pułapki – dodatkowe zagrożenie podczas potyczki. Zawierają nazwę, opis oraz poziom trudności.

\* potyczki – zawierają one nazwę oraz opis.

\* osoby – są to konta użytkowników. Zawierają nazwę, login oraz hasło.

## 6. Zaprojektowanie relacji pomiędzy encjami

Encja rasy jest połączona relacją N-M z encją tereny, gdyż jedna rasa może zamieszkiwać wiele terenów.

Encja lokacje jest połączona relacją N-1 z encją tereny, gdyż wiele lokacji może być na takim samym terenie.

Encja potwory jest połączona relacją N-1 z encją rasy, gdyż wiele potworów może wywodzić się z tej samej rasy.

Encja potyczki jest połączona relacjami N-M z encjami potwory, pułapki, skarby, gdyż może występować dowolna liczba tych elementów (lecz co najmniej jeden z każdego typu) oraz relacją N-1 z encją lokacje, gdyż każda potyczka musi mieć miejsce, gdzie się będzie odbywała, a także relacją N-1 z encją osoby, gdyż każda potyczka powinna mieć swojego autora.

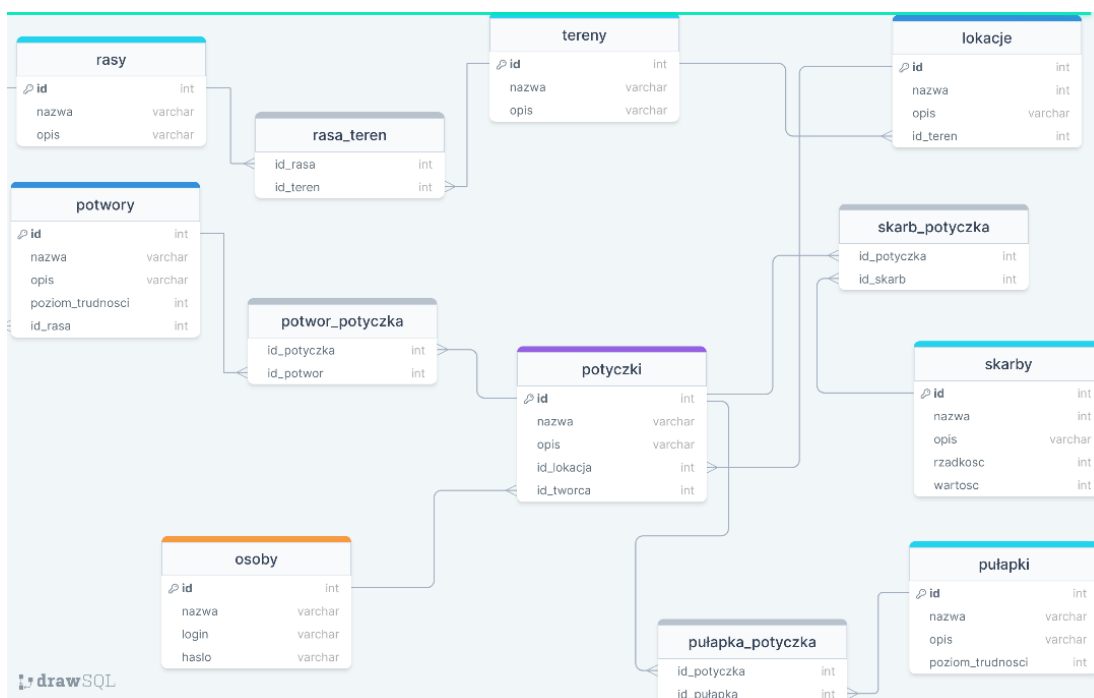


Diagram ERD projektu bazy danych

### III. Projekt logiczny

#### 7. Projektowanie tabel, kluczy, indeksów

Tworzenie tabel oraz ich łączenie zostało zdefiniowane w pliku *DDL.sql*.

#### 8. Słowniki danych

Tabela/Atrybut	Cel	Typ	Długość	Zakres	Unikalny?	Null?
potwory / id	Klucz główny	INTEGER	-	-	TAK	NIE
potwory / nazwa	Dane opisowe	VARCHAR	100	-	TAK	NIE
potwory / opis	Dane opisowe	VARCHAR	1000	-	NIE	NIE
potwory / poziom_trudnosci	Dane opisowe	INTEGER	-	0-10	NIE	NIE
potwory / id_rasa	Klucz obcy	INTEGER	-	-	NIE	NIE
tereny / id	Klucz główny	INTEGER	-	-	TAK	NIE
tereny / nazwa	Dane opisowe	VARCHAR	100	-	TAK	NIE
tereny / opis	Dane opisowe	VARCHAR	1000	-	NIE	NIE
potyczki / id	Klucz główny	INTEGER	-	-	TAK	NIE
potyczki / nazwa	Dane opisowe	VARCHAR	100	-	TAK	NIE
potyczki / opis	Dane opisowe	VARCHAR	1000	-	NIE	NIE
potyczki / id_lokacja	Klucz obcy	INTEGER	-	-	NIE	NIE
potyczki / id_tworca	Klucz obcy	INTEGER	-	-	NIE	TAK
sakrby / id	Klucz główny	INTEGER	-	-	TAK	NIE
skarby / nazwa	Dane opisowe	VARCHAR	100	-	TAK	NIE
skarby / opis	Dane opisowe	VARCHAR	1000	-	NIE	NIE
skarby / rzadkosc	Dane opisowe	VARCHAR	100	-	NIE	NIE
skarby / wartosc	Dane opisowe	INTEGER	-	-	NIE	NIE
rasy / id	Klucz główny	INTEGER	-	-	TAK	NIE
rasy / nazwa	Dane opisowe	VARCHAR	100	-	TAK	NIE
rasy / opis	Dane opisowe	VARCHAR	1000	-	NIE	NIE
lokacje / id	Klucz główny	INTEGER	-	-	TAK	NIE
lokacje / nazwa	Dane opisowe	VARCHAR	100	-	TAK	NIE
lokacje / opis	Dane opisowe	VARCHAR	1000	-	NIE	NIE
lokacje / id_teren	Klucz obcy	INTEGER	-	-	NIE	NIE
pulapki / id	Klucz główny	INTEGER	-	-	TAK	NIE
pulapki / nazwa	Dane opisowe	VARCHAR	100	-	TAK	NIE
pulapki / opis	Dane opisowe	VARCHAR	1000	-	NIE	NIE
pulapki / poziom_trudnosci	Dane opisowe	INTEGER	-	0-10	NIE	NIE
osoby / id	Klucz główny	INTEGER	-	-	TAK	NIE
osoby / nazwa	Dane opisowe	VARCHAR	100	-	TAK	NIE
osoby / login	Dane opisowe	VAECHAR	50	-	TAK	NIE
osoby / haslo	Dane opisowe	VARCHAR	50	-	NIE	NIE

## 9. Analiza zależności i normalizacja tabel

W bazie nie ma żadnych atrybutów, które zawierają wiele informacji na raz. Żadne atrybuty nie zależą funkcyjnie od żadnego klucza potencjalnego ani innych atrybutów niekluczowych ( rzadkość przedmiotu nie zależy od jego wartości, gdyż mały pierścień, nawet rzadki będzie mniej wart od dużego miecza ). Można stwierdzić, że baza danych znajduje się w 3NF.

## 10. ~~Denormalizacja struktury tabel [opcjonalne]~~

## 11. Zaprojektowanie operacji na danych

Funkcje używane w programie wraz z wyzwalaczami są zdefiniowane w pliku *DDL.sql*. Dodatkowo skrypt *sql\_utils.py* zawiera funkcje, które mają w sobie przygotowane polecenia SQL.

## **IV. Projekt funkcjonalny**

## 12. Interfejsy do prezentacji, edycji i obsługi danych

Na stronie głównej znajdują się przyciski, które przenoszą na strony, na których można albo zobaczyć wyniki, wprowadzić dane do tabeli lub wybrać element do usunięcia. Różne przyciski są widoczne dla osoby zalogowanej jako użytkownik, administrator oraz osoby niezalogowanej.

## 13. Wizualizacja danych

Dane wyświetlane są w postaci tabel. W tym celu zostały stworzone odpowiednie widoki. Dane związane z potyczkami można filtrować na podstawie wyliczonego poziomu trudności.

## 14. Zdefiniowanie panelu sterowania aplikacji

Aby skorzystać z aplikacji internetowej należy udać się na stronę główną. Tam znajdują się przyciski do wyświetlania, dodawania, a także usuwania elementów potyczek oraz samych potyczek. Dodatkowo są też przyciski do zalogowania się oraz rejestracji, a po zalogowaniu pojawia się przycisk do wylogowania. Po uzupełnieniu formularzy (każde pole należy wypełnić, a w przypadku wielokrotnego wyboru wybrać co najmniej jeden) należy nacisnąć odpowiedni przycisk. Poniżej znajduje się link do strony głównej. Wyświetlając potyczki mamy dostęp do możliwości filtrowania. Należy wybrać odpowiednie dane, a następnie

nacisnąć przycisk co powoduje przeładowanie wyników oraz wyświetlenie jedynie tych, które spełniają nadane przez użytkownika kryteria.

## 15. Makropolecenia

W aplikacji znajdują się funkcje do komunikacji z bazą danych (plik *sql\_utils.py*), natomiast w bazie danych zdefiniowane są funkcje oraz wyzwalacze, które pozwalają na łatwiejszy zapis do bazy danych.

## V. Dokumentacja

### 16. Wprowadzanie danych

Baza danych została uzupełniona przykładowymi danymi używając pliku *DML.sql*. Dalsze wprowadzanie danych odbywa się poprzez formularze występujące pod odpowiednimi adresami.

### 17. Dokumentacja użytkownika

Aby sprawdzić aplikację można skorzystać z linku: <https://django-server-production-0b52.up.railway.app/>

Aby uruchomić aplikację lokalnie należy mieć zainstalowanego lokalnie Pythona (zalecana wersja 3.10.8). Po rozpakowaniu plików należy udać się do katalogu z projektem i wpisać w terminal komendę:

```
pip install -r requirements.txt
```

Pobierze to potrzebne biblioteki. Następnie należy uruchomić aplikację używając

```
python manage.py migrate && python manage.py runserver
```

Aplikacja powinna uruchomić się pod adresem <http://127.0.0.1:8000/>. W celu załadowania plików DDL oraz DML i utworzenia bazy danych na nowo należy wpisać w adres link [http://127.0.0.1:8000/create\\_database](http://127.0.0.1:8000/create_database).

Aplikacja posiada administratora (login: ADMIN, hasło: ADMIN) oraz konto użytkownika (login: alan, hasło: alan). Na stronie głównej znajdują się odnośniki do wszystkich akcji możliwych do dokonania. Przy zapisywaniu potyczek należy pamiętać o zaznaczeniu opcji z każdej dziedziny (inaczej nie można zapisać potyczki).

### 18. Opracowanie dokumentacji technicznej

Dokumentacja funkcji do komunikacji z bazą danych znajduje się w pliku *sql\_utils.html*.

#### 19. Wykaz literatury

- \* <https://docs.djangoproject.com/en/4.1/topics/class-based-views/>
- \* <https://docs.djangoproject.com/en/4.1/topics/forms/>
- \* <https://www.postgresql.org/docs/current/>