

Projekt bazy danych "Słownika przyszłości"

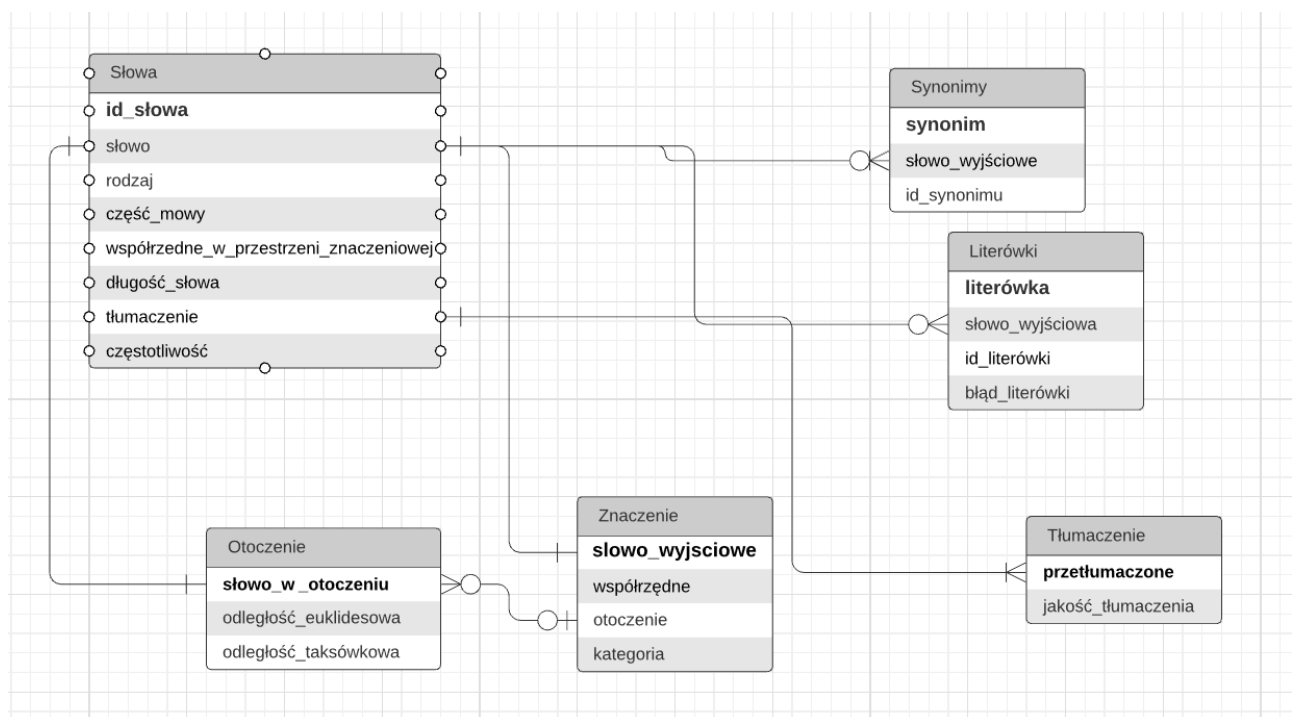
Michał Łopatka, Tomasz Gniazdowski

Bazy danych, Wtorek, 9:15-11:00

1 Wstęp - model konceptualny

Aplikacja bazodanowa "Słownika przyszłości" będzie mogła służyć jako słownik znaczeniowy dla modeli typu Word2Vec czy FastText, co pozwoli na większą wydajność dobierania odpowiednich wyrazów. Dodatkowo może stanowić wsparcie jako korekta słownikowa przy rozpoznawaniu tekstu na zeskanowanych dokumentach i obrazach oraz przy porównywaniu i analizie znaczeniowej całych zdań ze źródeł zewnętrznych. Dodatkową cechą będzie możliwość zwrócenia tłumaczenia danego słowa na zapytanie przy wtrąceniach obcojęzycznych.

2 Projekt schematu bazy danych



3 Opis świata rzeczywistego

Słowa w zdaniu statystycznie występują w otoczeniu zbioru słów, bądź ich synonimów, co może pozwolić na ich dopasowywanie, a także przybliżanie znaczenia całych zdań. Przy pomocy zbioru najczęstszych literówek oraz błędów ortograficznych, można z dużą dozą prawdopodobieństwa stwierdzić, że jest to szukany wyraz, nawet jeżeli nie został on zapisany poprawnie. Podczas automatycznego tłumaczenia słów, w przypadku nieuwzględnienia jego otoczenia może dochodzić do niejednoznaczności, szczególnie w przypadku homonimów. Uwzględnianie różnych znaczeń słowa może pomóc temu zaradzić.

4 Opracowanie wymagań funkcjonalnych i нефункциональных dla aplikacji bazodanowej

4.1 Wymagania funkcjonalne

4.1.1 Dodawanie oraz modyfikowanie słów słownika

Dodawaniem słów do słownika ma przede wszystkim zajmować się administrator serwisu. W tym zadaniu mają go jednak wesprzeć użytkownicy proponujący nowe wyrazy, które mogą zostać dodane oraz ewentualne modyfikacje już istniejących słów. Ponadto w przyszłości opracowany zostanie algorytm automatycznego tworzenia słownika na podstawie opracowanych modeli. Wszelkie modyfikacje użytkowników oraz opracowanego algorytmu muszą jednak zostać sprawdzone przez głównego administratora serwisu przed dodaniem ich do serwisu na stałe.

4.1.2 Usuwanie słów ze słownika

Jedynie administrator ma prawo do usuwania słów ze słownika.

4.1.3 Wyszukiwanie danych

Prawo do wyszukiwania danych słów o określonych kryteriach, ich sortowanie i tłumaczenie mają zarówno użytkownicy, jak i administrator systemu.

4.1.4 Sporządzanie statystyk i raportów

Prywatną historię ostatnich wyszukiń danej osoby wraz z datami posiada każdy użytkownik serwisu. Admin za to posiada dostęp do historii wszystkich wyszukiń serwisu wraz z niezbędnymi informacjami (takimi jak login użytkownika, który dane wyszukiwanie wywołał oraz datę wywołania). Na podstawie historii wyszukiń, ich ilości oraz częstości administrator może tworzyć raporty.

4.2 Wymagania нефункциональные

4.2.1 Tryby pracy

Aplikacja posiada graficzny tryb pracy. W zależności od przydzielonego trybu użytkownika (zwykły użytkownik lub admin) niektóre funkcje mogą zostać ograniczone.

4.2.2 Minimalne wymagania sprzętowe

Serwis musi być dostępny na komputerach wyposażonych w najbardziej popularne systemy operacyjne (Windows, MacOS, Linux). Ponadto komputery muszą posiadać (przynajmniej):

- Intel Core I3
- 500MB pamięci RAM

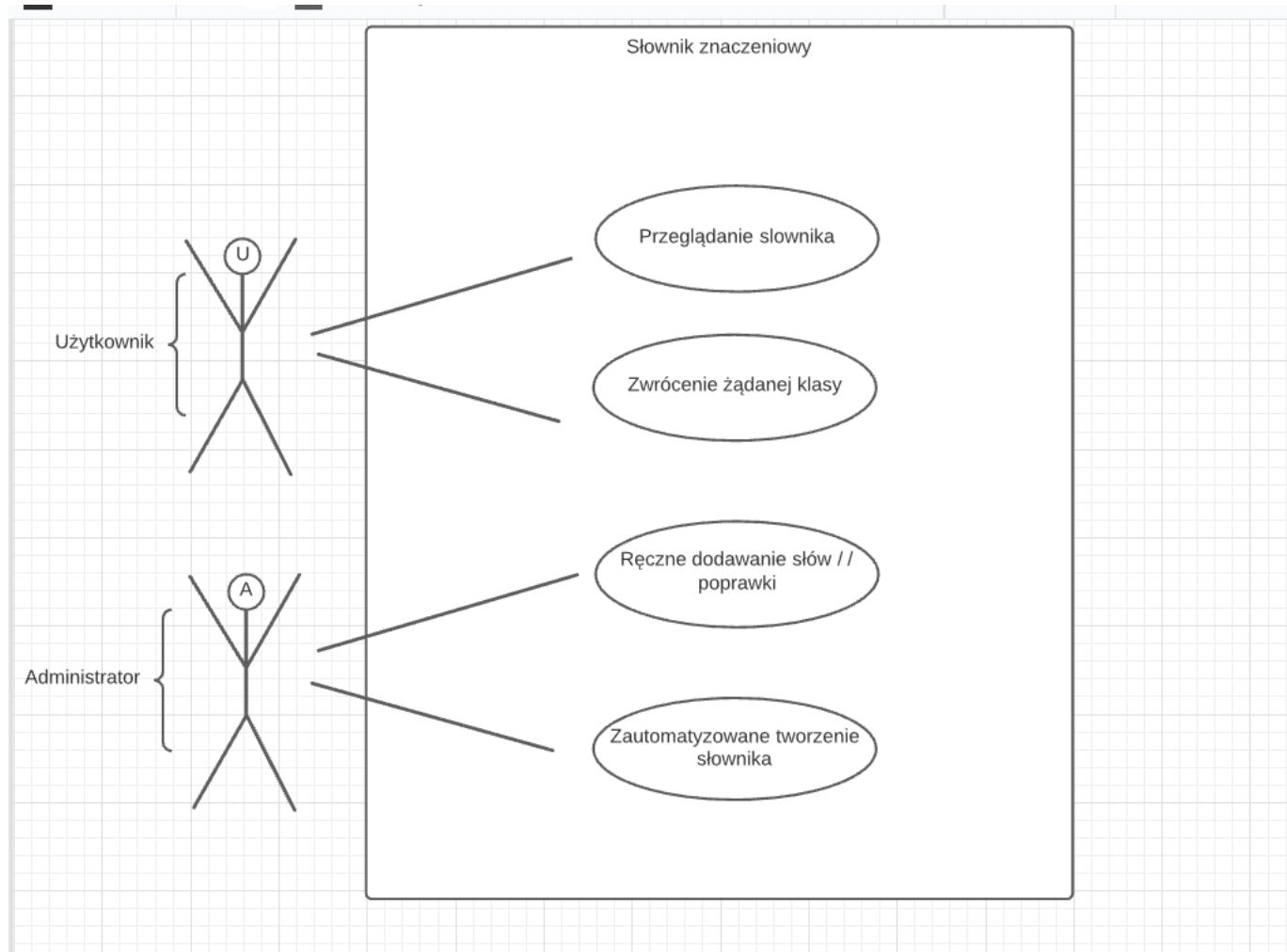
4.3 Środowisko implementacyjne oraz programistyczne

Środowiskiem implementacyjnym bazy danych jest MySQL, środowiskiem programistycznym Python.

4.4 Rodzaj bazy danych

Przedstawiona baza danych będzie relacyjną bazą danych.

5 Specyfikacja ogólnych wymagań funkcjonalnych za pomocą diagramu przypadków użycia



6 Analiza liczby instancji dla każdej encji

- Słowo
 - Liczba słów będzie systematycznie wzrastać, aż do poziomu kilku tysięcy najczęściej używanych wyrazów, pozostałe wyrazy w modelach będą mogły być tworzone łącząc te podstawowe.
- Synonimy
 - Liczebność synonimów jest ściśle powiązana z liczebnością wyrazów, będzie więc odpowiadająca.
- Literówki
 - Liczebność literówek będzie zależeć od ilości wyrazów, jednak może być ich kilku-krotnością.
- Tłumaczenia
 - Każdy wyraz będzie miał jedno lub więcej tłumaczeń, jednakże kilka wyrazów może mieć wspólne tłumaczenie. Dla słownika pełnego rozmiaru, ilość instancji nie powinna przekroczyć kilku tysięcy.

- Znaczenie
 - Każde słowo będzie mieć przypisane jedno znaczenie.
- Otoczenie
 - Ewentualne otoczenie będzie składać się z charakterystycznych słów, które dają nam jakieś informacje i są powtarzalne, ich ilość nie przekroczy tysiąca w słowniku docelowej wielkości.

7 Analiza użycia identyfikująca podstawowe rodzaje transakcji

7.1 Definicja aktorów

7.1.1 Użytkownik

Ma ograniczony dostęp do bazy. Użytkownik może wyszukiwać wyrazy zgodnie z określonym kryterium, sortować wyniki oraz je tłumaczyć. Dodatkowo może proponować modyfikacje i dodawanie nowych wyrazów. Modyfikacje takie muszą następnie zostać zaakceptowane przez admina, by na stałe znaleźć się w słowniku.

7.1.2 Admin

Ma pełny dostęp do wszystkich funkcji bazy. Jego głównym zadaniem jest dodawanie nowych rekordów (po ewentualnej konsultacji z ekspertami) oraz nadzorowanie działania algorytmu automatycznego tworzenia słownika. Ponadto admin ma za zadanie sprawdzania poprawności proponowanych nowych słów do słownika przez użytkowników oraz ich ewentualną modyfikację, akceptację lub odrzucanie.

7.2 Przypadki użycia

W każdym z poniższych przypadków użytkownik wprowadza określony wyraz w wyszukiwarkę słownika i wywołuje jedną z podanych poniżej funkcji. Ich wyniki wyświetlają się w odpowiednim miejscu na ekranie.

7.2.1 Wypisanie podstawowych informacji o wprowadzanym słowie

Aplikacja umożliwia wypisanie podstawowych informacji definiujących dane słowo (oraz modyfikację przez użytkownika które informacje o wyrazie mają zostać ukryte):

1. rodzaj gramatyczny (męski, żeński, nijaki)
2. część mowy (rzeczownik, czasownik, przymiotnik, liczebnik, przysłówek)
3. współrzędne w danej przestrzeni wyrazów (zależne od zastosowanego modelu i jego wymiarowości)
4. długość słowa (jako liczba naturalna, ilość liter z których składa się wyraz)
5. częstotliwość występowania słowa (określona w czasie uczenia modelu, określona w danej jednostce)
6. tłumaczenie słowa na dany język (zakłada się dwa tłumaczenia: język niemiecki oraz angielski)

7.2.2 Wypisanie słów spełniających określone kryteria, ich filtracja

Dzięki takiej ilości pól przyporządkowanych do każdego słowa możliwe jest również wypisywanie słów (oraz ich filtracja) zgodnie z określonymi przez użytkownika kryteriami (np. wszystkie rzeczowniki występujące w danej przestrzeni wyrazów).

7.2.3 Sortowanie słów zgodnie z określonym kryterium

Aplikacja umożliwia również sortowanie wyrazów na podstawie określonego kryterium (np. od słów najkrótszych do słów najdłuższych lub od słów najrzadziej występujących do tych występujących najczęściej w obrębie danego słowa).

7.2.4 Wypisanie najbliższego otoczenia danego słowa

Serwis umożliwia również wypisanie najbliższego otoczenia słowa, na przykład wypisanie pięciu wyrazów z otoczenia słowa *"Warszawa"* (przewiduje się również wybór normy jaka ma zostać użyta - taksówkowa lub euklidesowa oraz filtrację wyszukanych słów). Możliwe będzie dzięki temu wypisanie synonimów danego wyrazu, ponieważ słowa w zdaniu statystycznie występują w otoczeniu zbioru słów, bądź ich wyrazów bliskoznacznych, co może pozwolić na ich dopasowywanie, a także przybliżanie znaczenia całych zdań.

7.2.5 Poprawienie literówki, błędu ortograficznego we wprowadzanym słowie

Aplikacja po nieudanej próbie znalezienia wpisanego wyrazu musi przeszukać bazę najczęściej występujących literówek oraz błędów ortograficznych przypisanych do danych słów (posegregowanych od najczęściej do najrzadziej występujących, co może usprawnić działanie algorytmu). Przy pomocy zbioru najczęstszych literówek oraz błędów ortograficznych, można z dużą dozą prawdopodobieństwa stwierdzić, że jest to szukany wyraz, nawet jeżeli nie został on zapisany poprawnie. Po nieudanej próbie przeszukania częstych błędów w pisowni danego wyrazu serwis wyśle informacje o ewentualnym brakującym wyrazie do administratora.

7.2.6 Przetłumaczenie słowa, podanie jakości tłumaczenia

Aplikacja ma również dawać możliwość przetłumaczenia danego wyrazu na określony język oraz obliczenie jakości tego tłumaczenia (algorytm ocenieni, czy wyraz po przetłumaczeniu znajduje się w podobnej przestrzeni wyrazowej jak ten, przed tłumaczeniem). Wyeliminuje się w ten sposób występowanie błędnych tłumaczeń wynikających z występowania homonimów - wyrazów, które posiadają wiele znaczeń, występują one szczególnie często w języku angielskim.

8 Realizacja bazy danych

8.0.1 Wybrane narzędzia

Baza danych została stworzona jako model w MySQLWorkbench i wyeksportowana na server MySQL Server. Stworzono użytkowników o pełnym (administrator) i znacznie ograniczonym (zwykły użytkownik) dostępie do bazy danych. Połączenie z bazą można realizować z dowolnego komputera w sieci lokalnej.

8.0.2 Tworzenie bazy danych

Skrypt tworzący bazę danych:

```
1  -- MySQL Script generated by MySQL Workbench
2  -- Mon Apr 26 16:27:01 2021
3  -- Model: New Model Version: 1.0
4  -- MySQL Workbench Forward Engineering
5
6  SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
7  SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
8  SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,
9  NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
10
11  -- -----
12  -- Schema mydb
13  -- -----
14
15  -- -----
16  -- Schema mydb
17  -- -----
18  CREATE SCHEMA IF NOT EXISTS 'mydb' DEFAULT CHARACTER SET utf8 ;
19  USE 'mydb' ;
20
21  -- -----
22  -- Table 'mydb`.`Slowa`
23  -- -----
24  CREATE TABLE IF NOT EXISTS 'mydb`.`Slowa' (
25    'id_Slowa' INT(5) NOT NULL AUTO_INCREMENT,
26    'slowo' VARCHAR(45) NOT NULL,
27    'rodzaj' VARCHAR(20) NULL,
28    'czesc_mowy' VARCHAR(15) NOT NULL,
29    'wspolrzedne_w_przestrzeni_znaczeniowej' DOUBLE NOT NULL,
30    'dlugosc_slowa' INT(5) UNSIGNED NOT NULL,
31    'tlumaczenie' VARCHAR(45) NOT NULL,
32    'czestotliwosc' DOUBLE UNSIGNED NOT NULL,
33    PRIMARY KEY ('id_Slowa'),
34    UNIQUE INDEX 'slowo_UNIQUE' ('slowo' ASC) VISIBLE,
35    UNIQUE INDEX 'tlumaczenie_UNIQUE' ('tlumaczenie' ASC) VISIBLE)
36  ENGINE = InnoDB;
37
38
39  -- -----
40  -- Table 'mydb`.`Synonimy`
41  -- -----
42  CREATE TABLE IF NOT EXISTS 'mydb`.`Synonimy' (
43    'synonim' VARCHAR(45) NOT NULL,
44    'slowo_wyjsciowe' VARCHAR(45) NOT NULL,
45    'id_synonimu' INT(5) UNSIGNED NOT NULL,
46    PRIMARY KEY ('synonim'),
47    CONSTRAINT 'slowo_synonim'
48      FOREIGN KEY ('slowo_wyjsciowe')
49      REFERENCES 'mydb`.`Slowa' ('slowo')
50    ON DELETE NO ACTION
```

```

51     ON UPDATE NO ACTION)
52 ENGINE = InnoDB;
53
54
55 -----
56 -- Table 'mydb`.`Literowki`
57 -----
58 CREATE TABLE IF NOT EXISTS 'mydb`.`Literowki' (
59     'literowka' VARCHAR(45) NOT NULL,
60     'slowo_wyjsciowe' VARCHAR(45) NOT NULL,
61     'id_literowki' INT(6) NOT NULL,
62     'blad_literowki' DOUBLE UNSIGNED NOT NULL,
63     PRIMARY KEY ('literowka'),
64     INDEX 'slowo_literowka_idx' ('slowo_wyjsciowe' ASC) VISIBLE,
65     CONSTRAINT 'slowo_literowka'
66         FOREIGN KEY ('slowo_wyjsciowe')
67         REFERENCES 'mydb`.`Slova' ('slowo')
68         ON DELETE NO ACTION
69         ON UPDATE NO ACTION)
70 ENGINE = InnoDB;
71
72
73 -----
74 -- Table 'mydb`.`Tlumaczenie`
75 -----
76 CREATE TABLE IF NOT EXISTS 'mydb`.`Tlumaczenie' (
77     'przetlumaczone' VARCHAR(45) NOT NULL,
78     'jakosc_tlumaczenia' DOUBLE UNSIGNED NOT NULL,
79     PRIMARY KEY ('przetlumaczone'),
80     CONSTRAINT 'tlumaczenie_slowo'
81         FOREIGN KEY ('przetlumaczone')
82         REFERENCES 'mydb`.`Slova' ('tlumaczenie')
83         ON DELETE NO ACTION
84         ON UPDATE NO ACTION)
85 ENGINE = InnoDB;
86
87
88 -----
89 -- Table 'mydb`.`Otoczenie`
90 -----
91 CREATE TABLE IF NOT EXISTS 'mydb`.`Otoczenie' (
92     'slowo_w_otoczeniu' VARCHAR(45) NOT NULL,
93     'odleglosc_euklidesowa' DOUBLE UNSIGNED NOT NULL,
94     'odleglosc_taksowkowa' DOUBLE UNSIGNED NOT NULL,
95     PRIMARY KEY ('slowo_w_otoczeniu'),
96     CONSTRAINT 'otoczenie_slowo'
97         FOREIGN KEY ('slowo_w_otoczeniu')
98         REFERENCES 'mydb`.`Slova' ('slowo')
99         ON DELETE NO ACTION
100        ON UPDATE NO ACTION)
101 ENGINE = InnoDB;
102
103
104 -----
105 -- Table 'mydb`.`Znaczenie`
106 -----
107 CREATE TABLE IF NOT EXISTS 'mydb`.`Znaczenie' (
108     'slowo_wyjsciowe' VARCHAR(45) NOT NULL,
109     'wspolrzedne' DOUBLE NOT NULL,
110     'otoczenie' VARCHAR(45) NOT NULL,
111     'kategoria' VARCHAR(25) NOT NULL,

```

```

112 PRIMARY KEY ('slowo_wyjsciowe'),
113 INDEX 'otoczenie_znaczenie_idx' ('otoczenie' ASC) VISIBLE,
114 CONSTRAINT 'slowo_wsp'
115     FOREIGN KEY ('slowo_wyjsciowe')
116     REFERENCES 'mydb'.'Slowa' ('slowo')
117     ON DELETE NO ACTION
118     ON UPDATE NO ACTION,
119 CONSTRAINT 'otoczenie_znaczenie'
120     FOREIGN KEY ('otoczenie')
121     REFERENCES 'mydb'.'Otoczenie' ('slowo_w_otoczeniu')
122     ON DELETE NO ACTION
123     ON UPDATE NO ACTION)
124 ENGINE = InnoDB;
125
126 USE 'mydb' ;
127
128 -----
129 -- Placeholder table for view 'mydb'.'view1'
130 -----
131 CREATE TABLE IF NOT EXISTS 'mydb'.'view1' ('id' INT);
132
133 -----
134 -- routine1
135 -----
136
137 DELIMITER $$
138 USE 'mydb'$$
139 $$
140
141 DELIMITER ;
142
143 -----
144 -- View 'mydb'.'view1'
145 -----
146 DROP TABLE IF EXISTS 'mydb'.'view1';
147 USE 'mydb';
148
149
150 SET SQL_MODE=@OLD_SQL_MODE;
151 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
152 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

8.0.3 Wprowadzanie przykładowych danych

```
1 • INSERT INTO slowa(slowo, rodzaj, czesc_mowy, wspolrzedne_w_przestrzeni_znaczeniowej, dlugosc_slowa, tlumaczenie,
2   czestotliwosc) VALUES("komputer", "męski", "rzeczownik", 7.0, 8, "computer", 0.0013);
3 • INSERT INTO slowa(slowo, rodzaj, czesc_mowy, wspolrzedne_w_przestrzeni_znaczeniowej, dlugosc_slowa, tlumaczenie,
4   czestotliwosc) VALUES("klawiatura", "żeński", "rzeczownik", 8.0, 10, "keyboard", 0.0014);
5 • INSERT INTO slowa(slowo, rodzaj, czesc_mowy, wspolrzedne_w_przestrzeni_znaczeniowej, dlugosc_slowa, tlumaczenie,
6   czestotliwosc) VALUES("zasilacz", "męski", "rzeczownik", 3.0, 8, "power supply", 0.0002);
7 • INSERT INTO slowa(slowo, rodzaj, czesc_mowy, wspolrzedne_w_przestrzeni_znaczeniowej, dlugosc_slowa, tlumaczenie,
8   czestotliwosc) VALUES("program", "męski", "rzeczownik", 6.0, 7, "program", 0.0003);
9 • INSERT INTO slowa(slowo, rodzaj, czesc_mowy, wspolrzedne_w_przestrzeni_znaczeniowej, dlugosc_slowa, tlumaczenie,
10  czestotliwosc) VALUES("pomarańcza", "żeński", "rzeczownik", 26.0, 10, "orange", 0.00024);
11 • INSERT INTO slowa(slowo, rodzaj, czesc_mowy, wspolrzedne_w_przestrzeni_znaczeniowej, dlugosc_slowa, tlumaczenie,
12  czestotliwosc) VALUES("jabłko", "męski", "rzeczownik", 10.0, 6, "apple", 0.00022);
13 • INSERT INTO slowa(slowo, rodzaj, czesc_mowy, wspolrzedne_w_przestrzeni_znaczeniowej, dlugosc_slowa, tlumaczenie,
14  czestotliwosc) VALUES("ładny", "męski", "przymiotnik", 5.0, 5, "pretty", 0.00012);
15
16 • INSERT INTO literowki() VALUES("komputr", "komputer", 234, 0.1);
17 • INSERT INTO literowki() VALUES("klawiatora", "klawiatura", 4323, 0.15);
18 • INSERT INTO literowki() VALUES("zasilcz", "zasilacz", 534, 0.09);
19 • INSERT INTO literowki() VALUES("prgram", "program", 2432, 0.11);
20 • INSERT INTO literowki() VALUES("pomarancza", "pomarańcza", 463, 0.02);
21 • INSERT INTO literowki() VALUES("japko", "jabłko", 143, 0.7);
22
23 • INSERT INTO synonimy() VALUES("komputer", "PC", 2001);
24 • INSERT INTO synonimy() VALUES("klawiatura", "querty", 2002);
25 • INSERT INTO synonimy() VALUES("zasilacz", "bateryjka", 2003);
26 • INSERT INTO synonimy() VALUES("program", "aplikacja", 2004);
27 • INSERT INTO synonimy() VALUES("pomarańcza", "mandarynka", 2005);
28 • INSERT INTO synonimy() VALUES("jabłko", "papierówka", 2006);
29 • INSERT INTO synonimy() VALUES("ananas", "pajnapł", 2007);
30 • INSERT INTO synonimy() VALUES("ładny", "piękny", 2008);
31
32 • INSERT INTO tlumaczenie() VALUES("computer", 1.0);
33 • INSERT INTO tlumaczenie() VALUES("keyboard", 1.0);
34 • INSERT INTO tlumaczenie() VALUES("power supply", 1.0);
35 • INSERT INTO tlumaczenie() VALUES("program", 1.0);
36 • INSERT INTO tlumaczenie() VALUES("orange", 0.9);
37 • INSERT INTO tlumaczenie() VALUES("apple", 0.9);
38 • INSERT INTO tlumaczenie() VALUES("pretty", 0.97);
39
40 • insert into otoczenie() values("komputer", 2.34, 3.23);
41 • INSERT INTO otoczenie() VALUES("klawiatura", 2.34, 4.54);
42 • INSERT INTO otoczenie() VALUES("zasilacz", 5.84, 2.31);
43 • INSERT INTO otoczenie() VALUES("program", 3.23, 3.12);
44 • INSERT INTO otoczenie() VALUES("pomarańcza", 6.23, 1.31);
45 • INSERT INTO otoczenie() VALUES("jabłko", 2.42, 3.34);
46 • INSERT INTO otoczenie() VALUES("ładny", 0.6, 1.90);
47
48
49 • INSERT INTO znaczenie() VALUES("klawiatura", 1.0, "komputer", "IT");
50 • INSERT INTO znaczenie() VALUES("komputer", 5.0, "klawiatura", "IT");
51 • INSERT INTO znaczenie() VALUES("zasilacz", 4.0, "klawiatura", "elektronika");
52 • INSERT INTO znaczenie() VALUES("program", 17.0, "komputer", "IT");
53 • INSERT INTO znaczenie() VALUES("jabłko", 19.0, "pomarańcza", "owoce");
54 • INSERT INTO znaczenie() VALUES("pomarańcza", 12.0, "jabłko", "owoce");
55 • INSERT INTO znaczenie() VALUES("ładny", 21.0, "program", "wygląd");
```

8.0.4 Operacje grupowania, złączenia, agregujące

```
1  -- -----
2  -- Wyświetlenie informacji o jednym słowie z jednej tabeli
3  -- -----
4  select * from slowa
5  where slowa.slowo="zasilacz";
6
7  -- -----
8  -- Wyświetlenie informacji o jednym słowie z dwóch tabel
9  -- -----
10 select slowa.slowo, slowa.rodzaj, literowki.literowka, literowki.blad_literowki
11 from slowa
12 join literowki
13 on literowki.slowo_wyjsciowe = slowa.slowo
14 where slowa.slowo="zasilacz";
15
16 -- -----
17 -- Wyświetlenie informacji o słowach z dwóch tabel spełniających dany warunek
18 -- -----
19 select slowa.slowo, slowa.rodzaj, literowki.literowka, literowki.blad_literowki
20 from slowa
21 join literowki
22 on literowki.slowo_wyjsciowe = slowa.slowo
23 where slowa.dlugosc_slowa <=7;
24
25 -- -----
26 -- Wyświetlenie danego słowa wpisanego poprawnie, bądź z najprawdopodobniejszym
    bledem
27 -- -----
28 select slowa.slowo, slowa.rodzaj, literowki.literowka, literowki.blad_literowki
29 from slowa
30 join literowki
31 on literowki.slowo_wyjsciowe = slowa.slowo
32 where slowa.slowo = "pomarancza" or literowki.literowka = "pomarancza";
33
34 -- -----
35 -- Sortowanie całej tabeli alfabetycznie
36 -- -----
37 select slowa.slowo, slowa.rodzaj
38 from slowa
39 order by slowa.slowo;
40
41 -- -----
42 -- Liczenie słów o danym rodzaju w tabeli przy pomocy funkcji group by
43 -- -----
44 select count(slowa.slowo), slowa.rodzaj
45 from slowa
46 group by slowa.rodzaj;
47
48 -- -----
49 -- Wyświetlenie słów i ich literówek w kolejności alfabetycznej
50 -- -----
51 SELECT slowo FROM Slowa
52 UNION
53 SELECT literowka FROM literowki
54 ORDER BY slowo;
```

| id_Slowa | slowo | rodzaj | czesc_mowy | wspolrzedne_w_przestrzeni_znaczeniowej | dlugosc_slowa | tłumaczenie | czestotliwosc |
|----------|----------|--------|------------|--|---------------|--------------|---------------|
| 2 | zasilacz | męski | rzeczownik | 3 | 8 | power supply | 0.0002 |

Rysunek 1: Wyświetlenie informacji o jednym słowie z jednej tabeli

| | slowo | rodzaj | literowka | blad_literowki |
|---|----------|--------|-----------|----------------|
| ▶ | zasilacz | męski | zasilcz | 0.09 |

Rysunek 2: Wyświetlenie informacji o jednym słowie z dwóch tabel

| | slowo | rodzaj | literowka | blad_literowki |
|---|---------|--------|-----------|----------------|
| ▶ | program | męski | prgram | 0.11 |
| | jabłko | męski | japko | 0.7 |
| | ładny | męski | ladny | 0.04 |

Rysunek 3: Wyświetlenie informacji o słowach z dwóch tabel spełniających dany warunek

| | slowo | rodzaj | literowka | blad_literowki |
|---|------------|--------|------------|----------------|
| ▶ | pomarańcza | żeński | pomarancza | 0.02 |

Rysunek 4: Wyświetlenie danego słowa wpisanego poprawnie, bądź z najprawdopodobniejszym błędem

| | slowo | rodzaj |
|---|------------|--------|
| ▶ | jabłko | męski |
| | klawiatura | żeński |
| | komputer | męski |
| | pomarańcza | żeński |
| | program | męski |
| | zasilacz | męski |
| | ładny | męski |

Rysunek 5: Sortowanie całej tabeli alfabetycznie

| | count(slowa.slowo) | rodzaj |
|---|--------------------|--------|
| ▶ | 2 | żeński |
| | 5 | męski |

Rysunek 6: Liczenie słów o danym rodzaju w tabeli przy pomocy funkcji group by

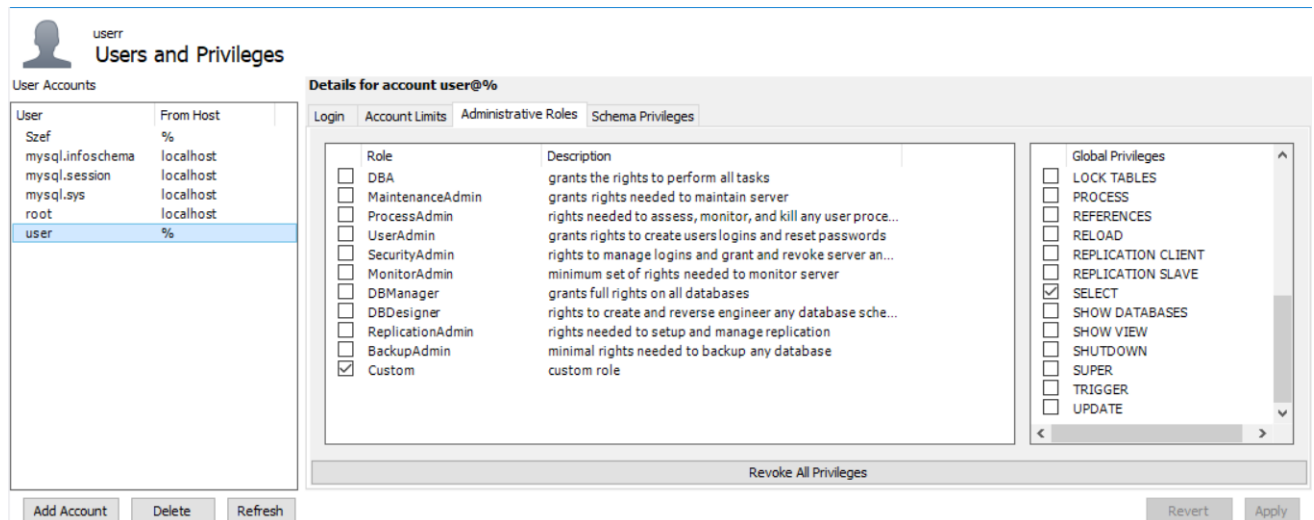
| | slowo |
|--|------------|
| | klawiatóra |
| | klawiatura |
| | komputer |
| | komputr |
| | ładny |
| | pomarańcza |
| | prgram |
| | program |
| | zasilacz |
| | zasilcz |
| | ładny |

Rysunek 7: Wyświetlenie słów i ich literówek w kolejności alfabetycznej

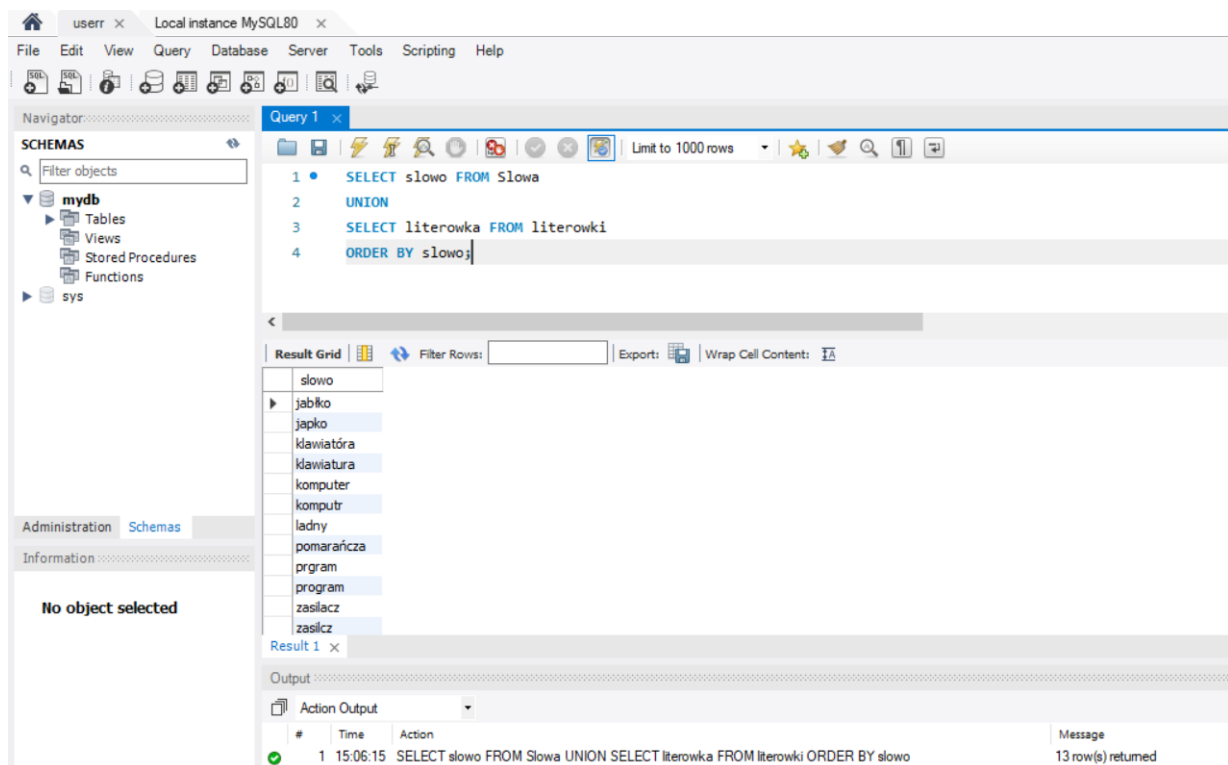
9 Polityka bezpieczeństwa

9.1 Zwykły użytkownik

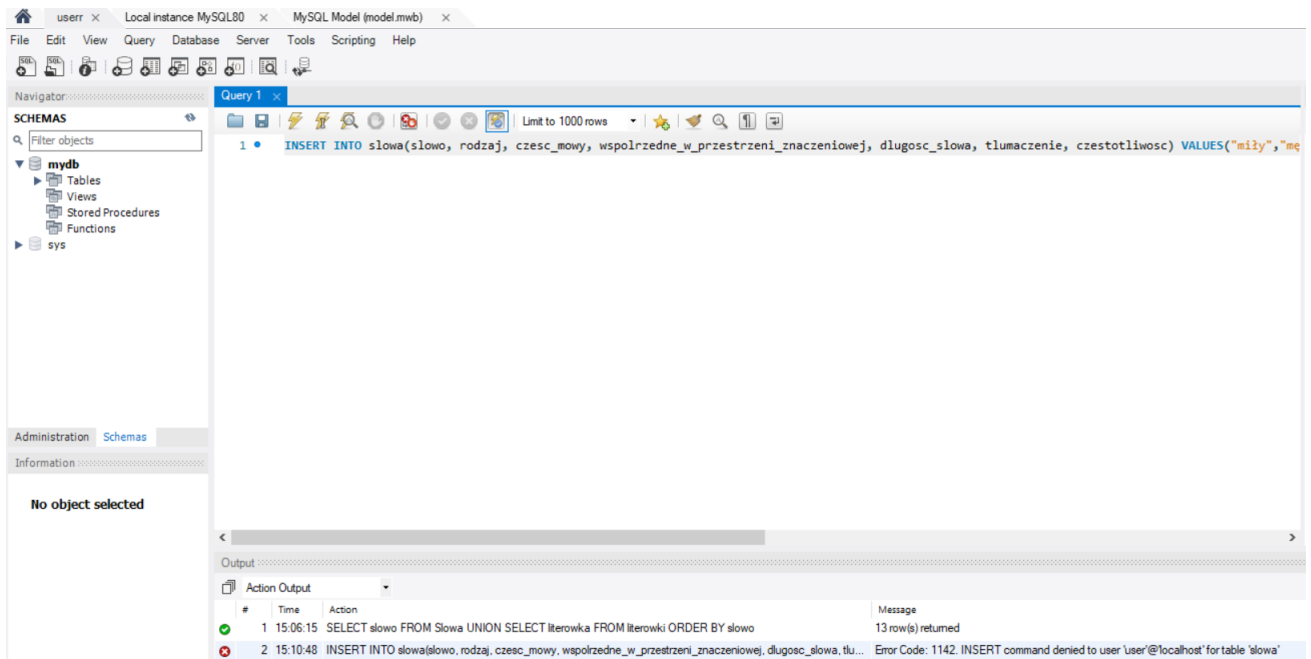
Zwykły użytkownik nie ma uprawnień do modyfikowania bazy, ma jedynie dostęp SELECT



Rysunek 8: Uprawnienia zwykłego użytkownika



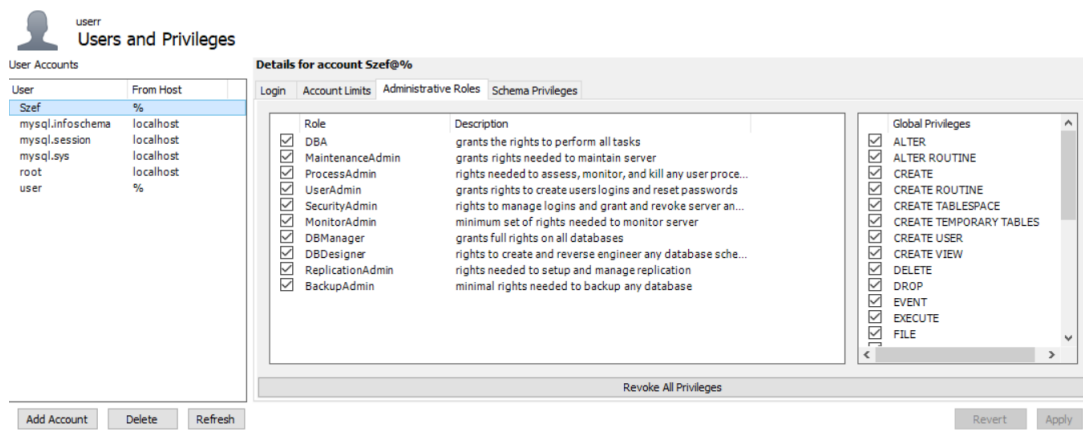
Rysunek 9: Zwykły użytkownik ma jedynie prawo do wyświetlania elementów bazy z określonymi kryteriami



Rysunek 10: Zwykły użytkownik nie ma prawa do modyfikowania elementów bazy

9.2 Administrator

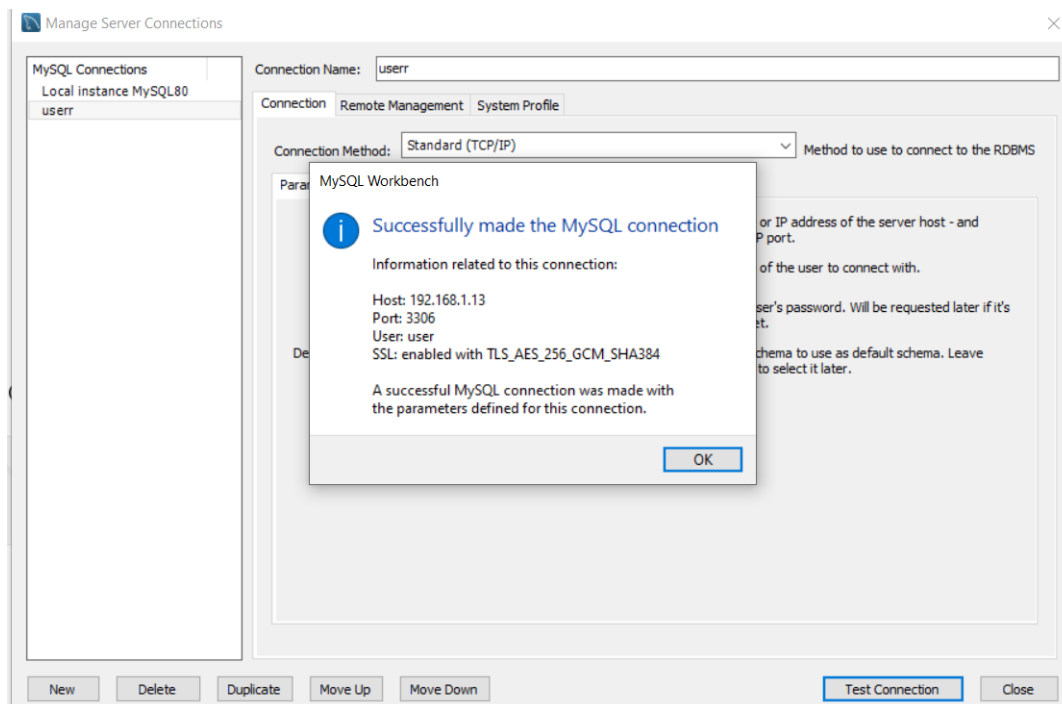
Administrator ma pełne prawo do modyfikowania elementów bazy danych.



Rysunek 11: Uprawnienia administratora bazy danych

9.3 Połączenie

Aby zminimalizować zagrożenia, użyto protokołu SSL do połączenia z serwerem bazy danych znajdującym się w sieci lokalnej.



10 Aplikacja bazodanowa

10.1 Realizacja aplikacji

Realizacja aplikacji została oparta na *Django* jako webowa aplikacja. Do połączenia z bazą danych *MySQL* niezbędny był sterownik API bazy danych - *mysqlclient*. Dodatkowo, niezbędne było wpisanie danych potrzebnych do standardowego połączenia.

10.2 Struktura Aplikacji

Modele przechowują odpowiednie pola klas, które potem poprzez widoki będą wywoływane na stronie napisanej w *html*. W momencie tworzeniu strony, bądź wpisania szukanego słowa, odpowiedni parametr jest przekazywany do widoków, które wywołują odpowiednie zapytanie *SQL*. Właściwe dane są zwracane i przypisywane do odpowiednich pól nowo utworzonego obiektu klasy. Te są przekazywane do wyświetlenia na żądanej stronie.

10.3 Zaimplementowane funkcjonalności

10.3.1 Menu główne

Pozwala wprowadzać szukane słowa, również z najczęstszymi literówkami, a także przejść do pozostałych podstron.

10.3.2 Informacje o słowie

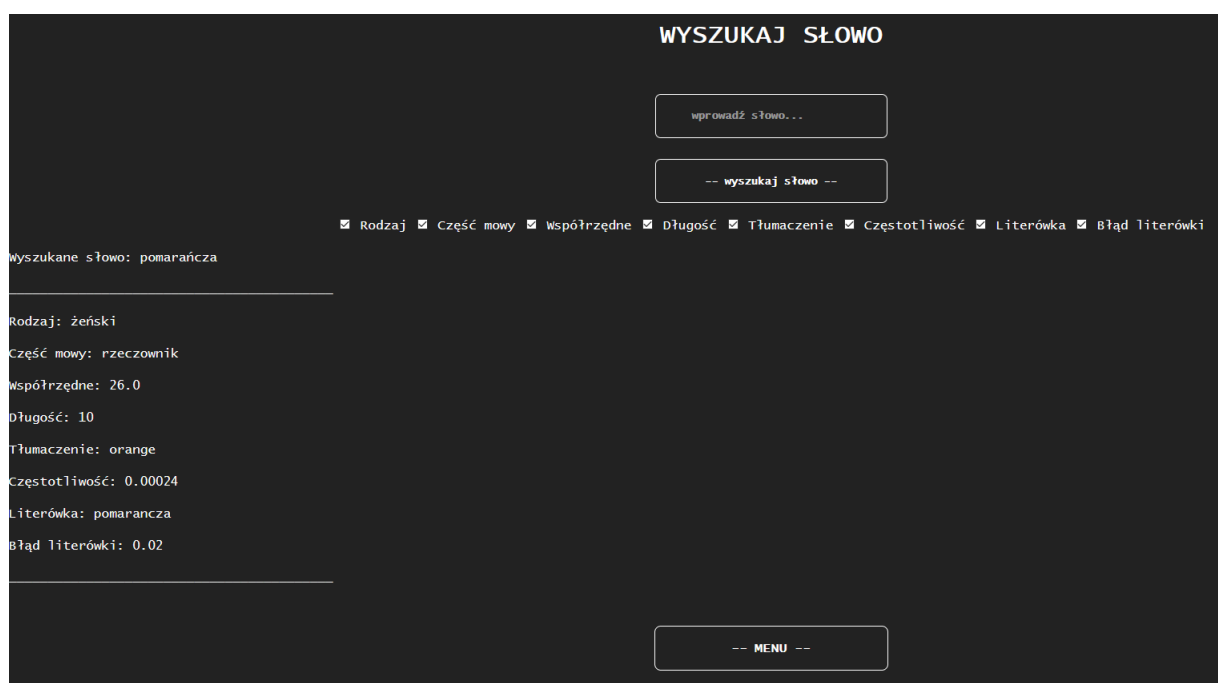
Po wyszukaniu, wyświetlają się pełne informacje o danym słowie, istnieje możliwość schowania niepotrzebnych danych.

10.3.3 Statystyki - liczba słów danego rodzaju

Umożliwia zobaczenie podliczonej sumy wszystkich wyrazów z bazy danego rodzaju.



Rysunek 12: Menu główne aplikacji.



Rysunek 13: Pełna informacja o wprowadzonym słowie.

10.3.4 Wszystkie wyrazy

Po przejściu na tę podstronę, widoczne są wszystkie wyrazy z bazy danych, wraz z ich rodzajami.

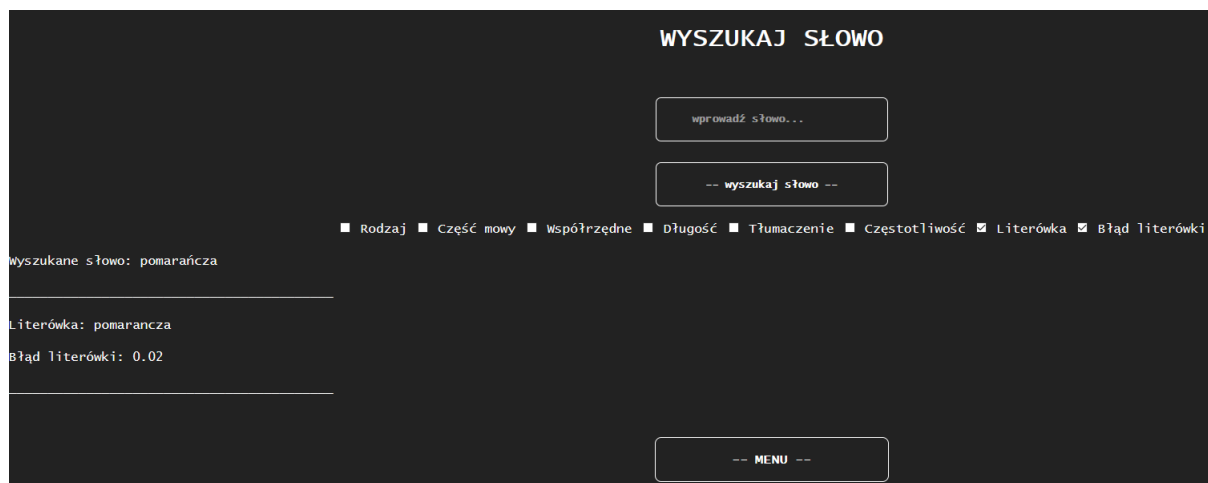
10.4 Bezpieczeństwo i optymalizacja

10.4.1 SQL Injection

Parametry nie są przekazywane do surowych zapytań *SQL* bezpośrednio, a poprzez słownik, co zapobiega wpisaniu słów kluczowych i ewentualnym atakom.

10.4.2 Komunikaty błędów

Informacje z debuggera zostały ukryte, gdy odpowiedni wyraz nie został znaleziony, bądź strona nie może zostać załadowana. Zwiększa to bezpieczeństwo aplikacji. Niepowołane osoby nie mogą mieć dostępu do takich danych, gdyż mogą zostać one użyte przy atakach.



Rysunek 14: Częściowa informacja o wprowadzonym słowie.



Rysunek 15: Statystyki słownika.



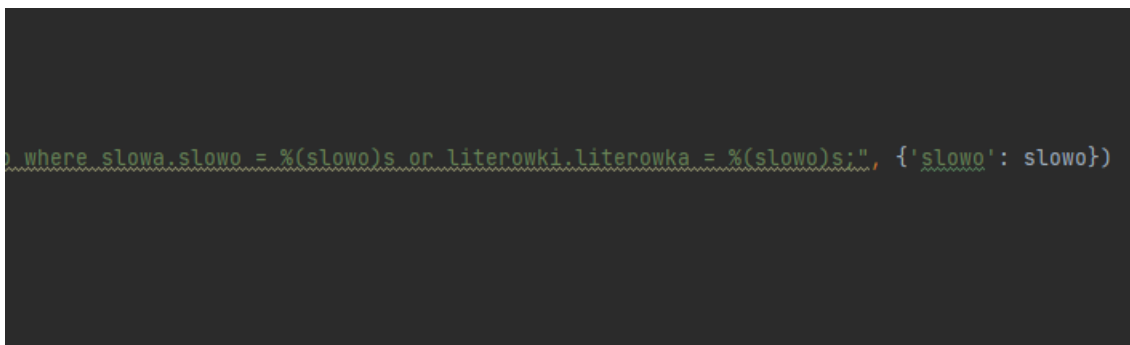
Rysunek 16: Wypisane wszystkie słowa słownika posegregowane alfabetycznie.

10.4.3 Brak tworzenia powtórnych połączeń

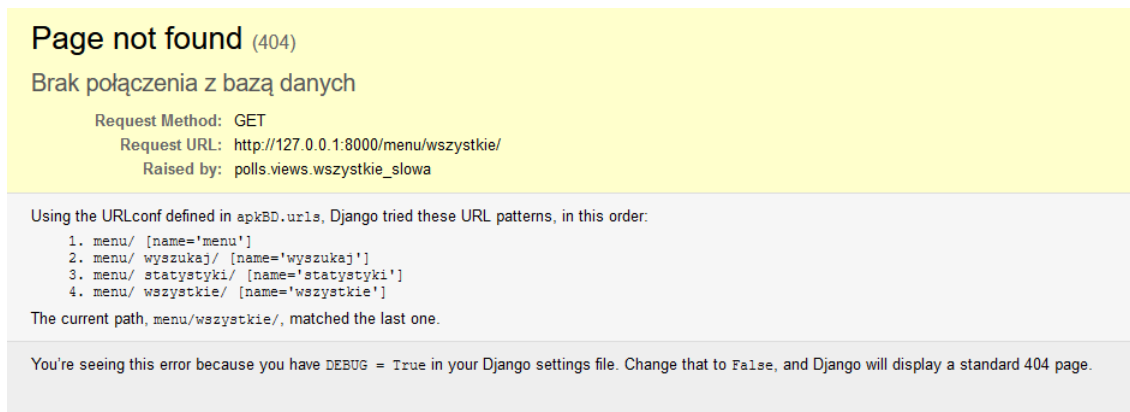
Zamiast tworzyć połączenie z bazą danym przy każdorazowym zapytaniu, tworzone jest jedno połączenie z bazą danych. Zwiększa to płynność działania aplikacji.

10.4.4 Backup

W programie *MySQL Workbench* przetestowany został backup wszystkich tabel z możliwością zapisania na osobnym dysku. Praktyka taka jest niezbędna, gdyż umożliwia szybkie odtworzenie bazy po ewentualnej awarii.



Rysunek 17: Sposób na zapobiegnięcie *SQL Injection*.



Rysunek 18: Komunikat o braku połączenia z bazą danych - niebezpieczne rozwiązanie.

Not Found

The requested resource was not found on this server.

Rysunek 19: Komunikat o braku połączenia z bazą danych - bezpieczne rozwiązanie.

10.4.5 Dodawanie użytkowników z poziomu administratora

Przetestowano dodawanie użytkowników w warstwie aplikacji bazodanowej.