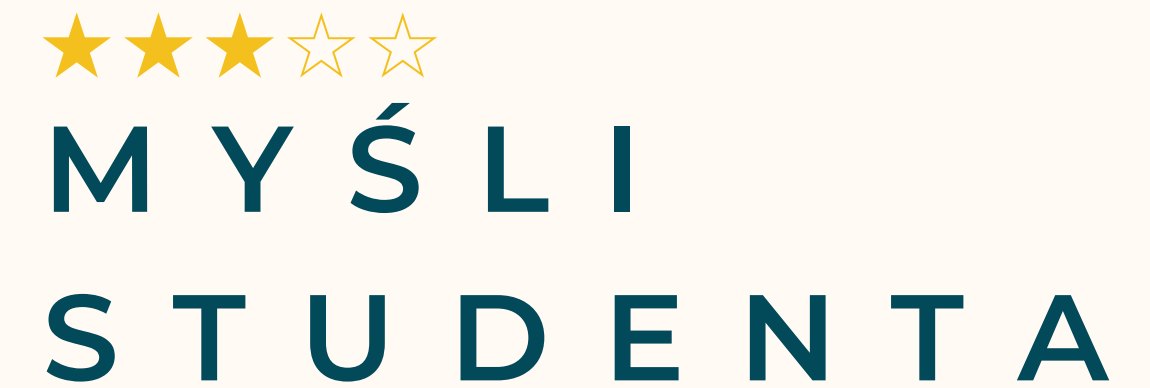




# MYŚLI STUDENTA

Czerwiec 2023

# SPIIS TREŚCI



## MYŚLI STUDENTA

- Nasze inspiracje i wizje
- Implementacja
- Testowanie, bezpieczeństwo i dobre praktyki
- Nasz zespół
- OD ZERA DO PYTHON DEVELOPERA



[Link do filmiku](#)

# Nasza wizja

- Stworzenie centralnego miejsca z informacjami dotyczącymi różnych kierunków studiów.

- Umożliwienie łatwego dostępu do liczby semestrów, tytułu naukowego, sylabusu i opinii studentów.

Grupa docelowa to maturzyści oraz studenci poszukujący informacji na temat różnych kierunków studiów.

- Weryfikacja studentów dzięki połączeniu z systemem USOS, co pozwala na bardziej wiarygodne opinie i odpowiedzi na pytania.

- Wymagania: strona internetowa oparta na frameworku Flask, baza danych, przeglądanie kierunków według miast, uczelni, formularze opinii i pytań, system moderacji treści.

# Wybór tematu

1

Dostęp do rzetelnych informacji na temat kierunków studiów może być wyzwaniem dla studentów.

2

Połączenie z systemem USOS dodaje element wiarygodności i weryfikacji informacji.

3

Stworzenie platformy, która ułatwi im podjęcie decyzji na podstawie opinii innych studentów, było dla nas naturalnym wyborem.

4

W rezultacie powstaje narzędzie dla studentów, pomagające im w znalezieniu odpowiednich informacji i podjęciu świadomej decyzji.

# Potencjał biznesowy

Duże  
zapotrzebowanie  
na źródło  
informacji na  
temat kierunków  
studiów.

Nasza strona  
internetowa  
oferuje unikalną  
wartość dodaną,  
umożliwiając  
łatwy dostęp do  
takich informacji  
w jednym miejscu

Wysokie  
zainteresowanie  
studentów i  
maturzystów ze  
względu na brak  
takiej platformy w  
Polsce

Nasz projekt ma  
budować zaufanie  
wśród  
użytkowników i  
oferować rzetelne  
informacje.



# GRUPA DOCELOWA

## Grupa docelowa



Nasza strona internetowa jest adresowana głównie do uczniów szkół średnich i studentów.

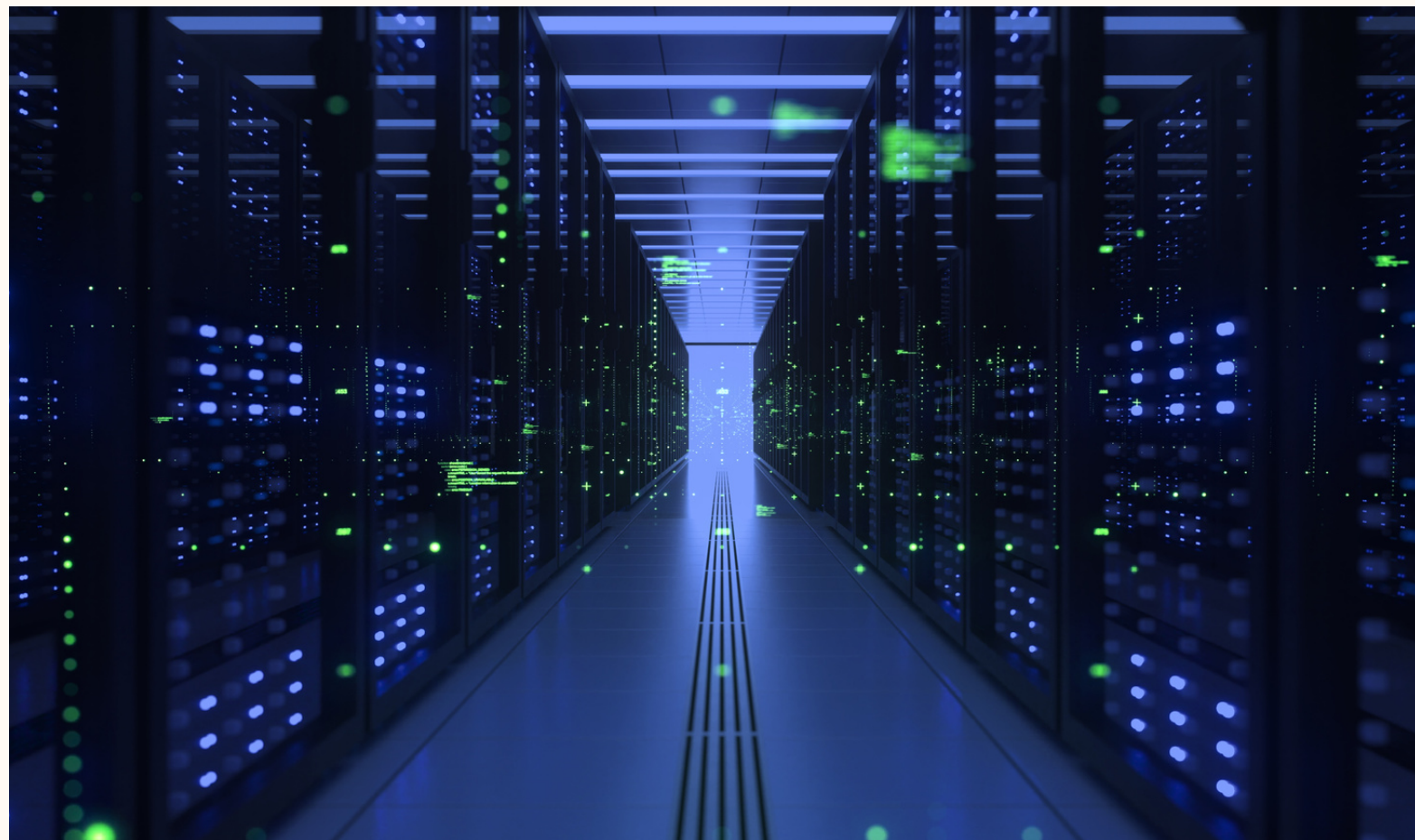
Jest to przydatne również dla osób niezadowolonych ze swoich studiów, rozważających zmianę kierunku.

Studenci poszukujący rzetelnych informacji na temat różnych kierunków studiów np. wybierając uczelnię i kierunek na drugi stopień .

Strona internetowa oferuje niezależne opinie studentów, co pomaga w podjęciu świadomej decyzji dotyczącej edukacji każdego młodego człowieka.

Strona jest także skierowana do absolwentów, aby pozyskać od nich opinię na temat kierunków, które ukończyli.

# BAZA DANYCH



- 01 Baza danych przechowuje informacje o kierunkach studiów, opinie studentów oraz inne istotne dane.
- 02 Umożliwia przeglądanie kierunków według miast, uczelni.
- 03 Integracja z systemem USOS umożliwia weryfikację opinii studentów.
- 04 Formularze do dodawania nowych opinii i pytań są również zapisywane w bazie danych.
- 05 System odpowiedzi na pytania oraz moderacji treści zapewniają jakość informacji na stronie.




# DODAWANIE OPINII

- 1 Formularze na naszej stronie umożliwiają dodanie nowych opinii oraz pytań.
- 2 Umożliwiają one studentom dzielenie się swoimi doświadczeniami z innymi użytkownikami.
- 3 Opinie są moderowane przez system odpowiedzi na pytania i moderacji treści.
- 4 Oceny i opinie pozwalają na uzyskanie rzetelnej informacji o danym kierunku studiów.
- 5 System pozwala również na weryfikację opinii, co zwiększa wiarygodność platformy.

[Wyloguj się](#) [Strona Główna](#) [Porównaj](#) [Uniwersytety](#) [O nas](#)

## Dodaj opinie

Opinia:

Trudność: 5 - bardzo trudne 

Jakość nauczania: 5 - fantastyczna 

---

**01**

Integracja z systemem USOS umożliwia weryfikację opinii.

---

---

**02**

Dzięki temu opinie są bardziej wiarygodne, zapobiega to fałszerstwom.

---

---

**03**

Użytkownicy mogą uwierzytelniać swoje opinie za pomocą logowania do systemu USOS.

---

**04**

Integracja z systemem USOS ułatwia również zbieranie informacji o użytkownikach i ich preferencjach w przyszłości.

---

**05**

To z kolei pozwala na lepsze personalizowanie oferty i udzielanie indywidualnych odpowiedzi na pytania w przyszłości.



# Integracja z systemem USOS

(dotyczy uczelni korzystających z systemu USOS)

# Przeglądanie kierunków studiów



Strona umożliwia przeglądanie kierunków według miast oraz uczelni.



Użytkownicy mogą łatwo znaleźć informacje o kierunku, który ich interesuje.



Można również przeglądać opinie dotyczące danego kierunku.



Możliwość filtrowania pozwala na dokładne dostosowanie wyników wyszukiwania.

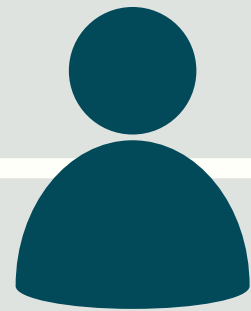


To umożliwia znalezienie odpowiednich informacji oraz ogólne zapoznanie się z dostępnymi kierunkami w danym mieście czy uczelni.

# Niezarejestrowany użytkownik powinien mieć możliwość:



MOŻLIWOŚĆ  
WYSZUKANIA  
KIERUNKÓW Z  
CAŁEJ POLSKI



SPRAWDZENIE  
DOSTĘPNYCH  
KIERUNKÓW NA  
DANEJ UCZELNI



SPRAWDZENIE  
OPINII O  
INTERESUJĄCYM  
KIERUNKU



WYŚWIETLENIE  
RANKINGÓW

# Zalogowany użytkownik powinien mieć możliwość:



TE SAME  
UPRAWNIENIA  
CO  
NIEZALOGOWANY



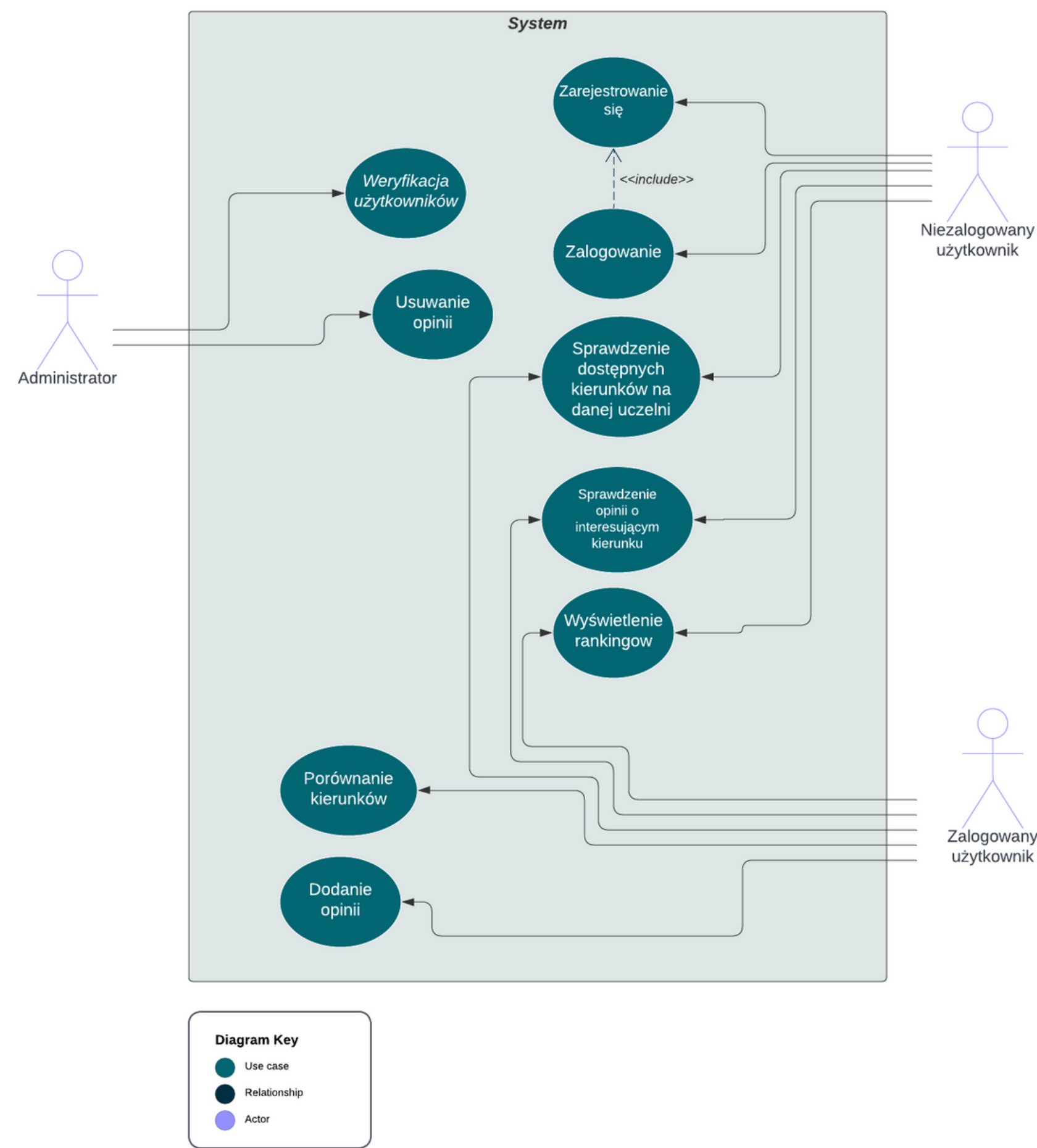
MOŻLIWOŚĆ  
DODANIA OPINII



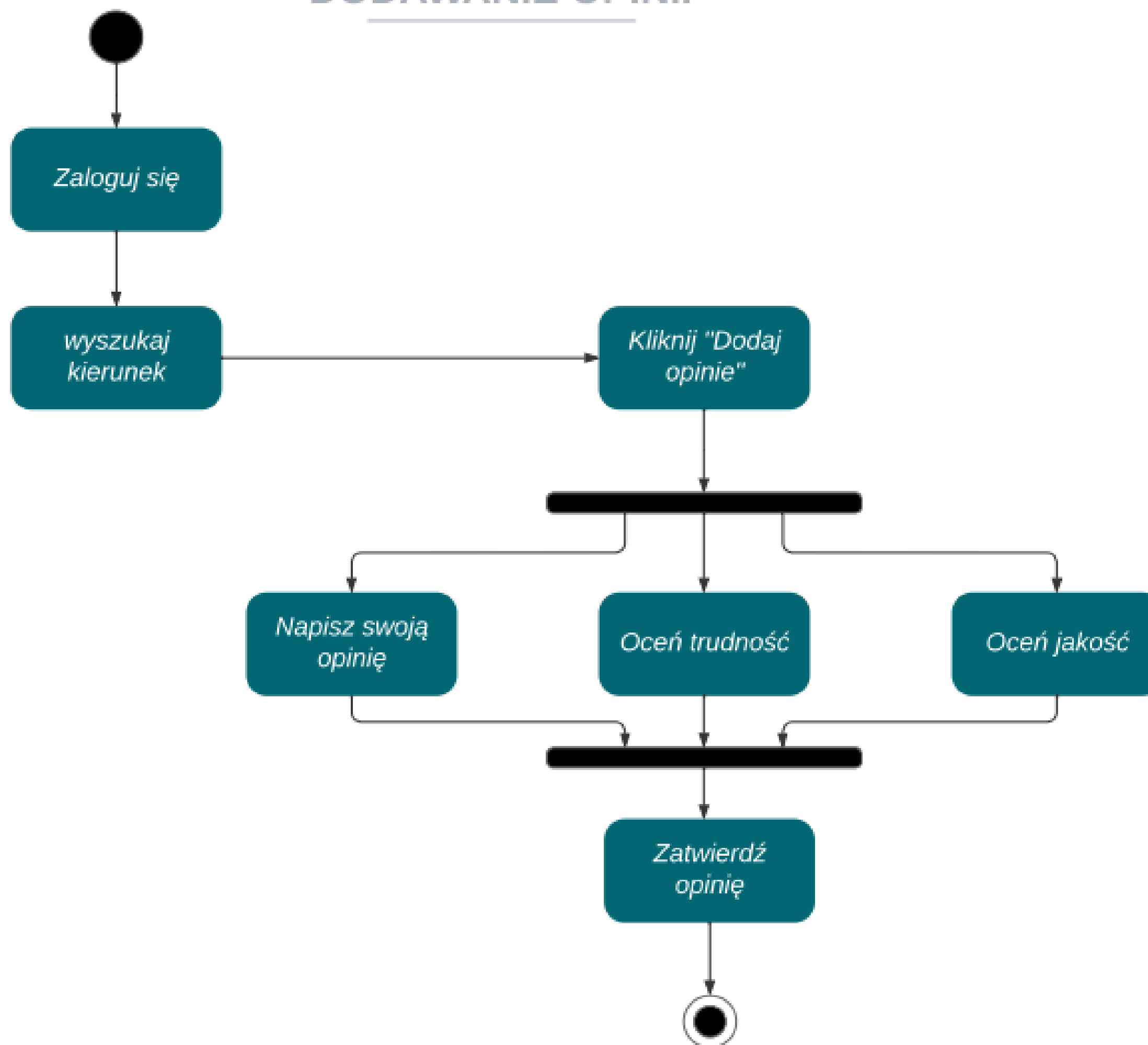
MOŻLIWOŚĆ  
PORÓWNAŃ  
KIERUNKÓW



Diagram przypadków użycia programu "Myśli studenta"



## DODAWANIE OPINII



# IMPLEMENTACJA



# Drzewo katalogów

```
├── university-review-system-main
│   ├── README.md
│   ├── instance
│   │   ├── insertdata.sql
│   │   └── unidatabase.db
│   ├── main.py
│   └── website
│       ├── __init__.py
│       ├── __pycache__
│       │   ├── __init__.cpython-310.pyc
│       │   ├── auth.cpython-310.pyc
│       │   ├── functions.cpython-310.pyc
│       │   ├── models.cpython-310.pyc
│       │   └── views.cpython-310.pyc
│       ├── admin.py
│       ├── auth.py
│       ├── functions.py
│       ├── models.py
│       ├── static
│       │   ├── 2.svg
│       │   └── style.css
│       ├── templates
│       │   ├── about.html
│       │   ├── addopinion.html
│       │   ├── base.html
│       │   ├── compare_courses.html
│       │   ├── course.html
│       │   ├── home.html
│       │   ├── index.html
│       │   ├── login.html
│       │   ├── sign_up.html
│       │   ├── uni-list.html
│       │   └── university.html
│       └── views.py
7 directories, 28 files
```

# PANEL ADMIN

Ten kod definiuje niestandardową klasę widoku (AdminView) dla Flask-Admin, która dziedziczy po ModelView z modułu flask\_admin.contrib.sqla. ModelView jest używany w Flask-Admin do automatycznego generowania interfejsu administracyjnego dla modeli bazodanowych.

```
1  from flask_admin.contrib.sqla import ModelView
2  from flask import session, redirect, url_for, request
3
4
5  class AdminView(ModelView):
6
7      def __init__(self, *args, **kwargs):
8          super().__init__(*args, **kwargs)
9          self.static_folder = 'static'
10
11      def is_accessible(self):
12          return session.get('user') == 'Administrator'
13
14      def inaccessible_callback(self, name, **kwargs):
15          if not self.is_accessible():
16              return redirect(url_for('home', next=request.url))
```



# PANEL ADMIN

W klasie AdminView są zdefiniowane trzy metody:

1

**`_init_(self, *args, **kwargs)`**

Jest to konstruktor klasy AdminView. Przesłanie tej metody pozwala na dodatkową konfigurację widoku. W tym przypadku, w konstruktorze ustawiane jest pole `static_folder` na wartość `'static'`. Może to być użyteczne, gdy chcemy skonfigurować folder statyczny dla plików CSS, JS itp.

2

**`is_accessible(self)`**

Jest to metoda, która decyduje, czy dany użytkownik ma dostęp do widoku administracyjnego. W tym przypadku, sprawdzane jest, czy w sesji użytkownika (przechowywanej w `session`) istnieje klucz `'user'`, którego wartość jest równa `'Administrator'`. Jeśli tak, zwraca `True`, co oznacza, że użytkownik ma dostęp. W przeciwnym razie zwraca `False`.

3

**`inaccessible_callback(self, name, **kwargs)`**

Jest to metoda, która jest wywoływana, gdy użytkownik nie ma dostępu do widoku administracyjnego. W tym przypadku, jeśli użytkownik nie ma dostępu, zostaje przekierowany na stronę główną (`home`) z dodatkowym parametrem `next`, który zawiera adres URL aktualnego żądania.

# PANEL ADMIN

W klasie AdminView są zdefiniowane trzy metody:

1

`_init_(self, *args, **kwargs)`

2

`is_accessible(self)`

3

`inaccessible_callback(self, name, **kwargs)`

Ten kod tworzy niestandardowy widok administracyjny, który ma ograniczony dostęp dla użytkowników, których wartość klucza 'user' w sesji jest ustawiona na 'Administrator'. Jeśli użytkownik nie ma dostępu, zostaje przekierowany na stronę główną.

# Framework FLASK



# Flask

- Strona internetowa oparta jest na frameworku Flask.
- Pozwala to na łatwe tworzenie i utrzymywanie strony internetowej.
- Flask charakteryzuje się dobrym skalowaniem i wydajnością.
- Dzięki temu użytkownicy mogą szybko i sprawnie przeglądać strony i dodawać swoje opinie.
- Framework Flask jest również dobrze udokumentowany, co ułatwia pracę z nim.
- Flask-SQLAlchemy to rozszerzenie dla frameworka Flask, które zapewnia integrację z bazą danych za pomocą SQLAlchemy.

# SQLite



## Lekkość

SQLite jest lekką bazą danych, która nie wymaga zewnętrznego serwera bazy danych do działania.

Baza danych jest przechowywana w jednym pliku, co ułatwia jej przenoszenie i zarządzanie.



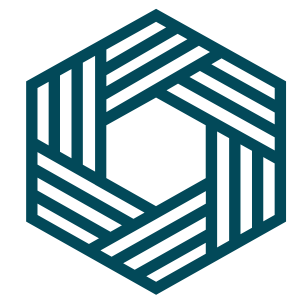
## Wydajność

SQLite ma dobre osiągi i może obsługiwać duże ilości danych. Ze względu na swoją lekkość i bez potrzeby połączenia z zewnętrznym serwerem, SQLite działa szybko i efektywnie.



## Dokumentacja

SQLite jest dobrze udokumentowane. Istnieje wiele przykładów, poradników i dokumentacji, które pomagają w pracy z tą bazą danych.



Dzięki wbudowanej obsłudze SQLite w Flasku, można łatwo tworzyć i zarządzać bazami danych SQLite w aplikacji opartej na Flasku.

```

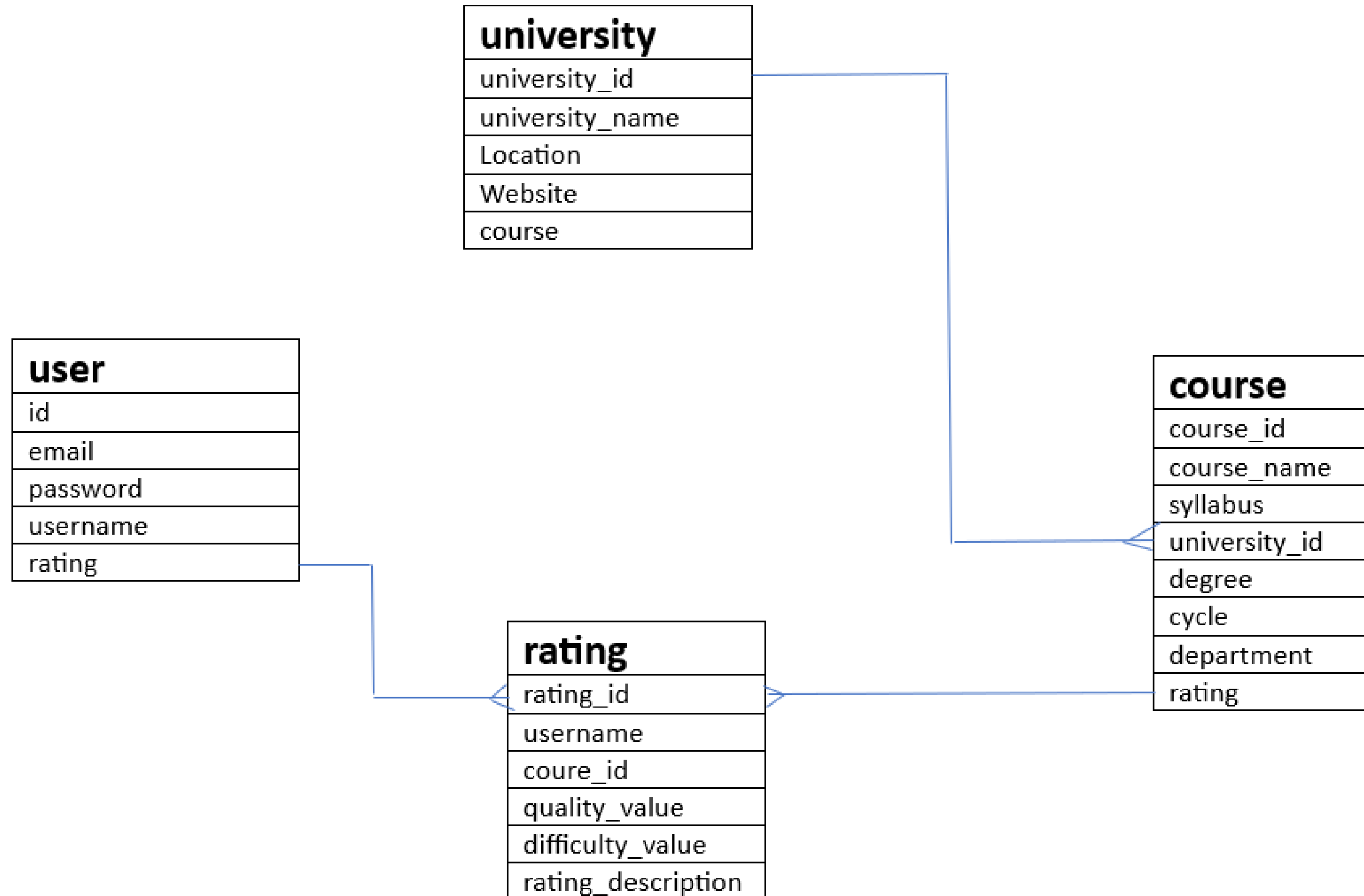
68 @views.route('/_update_course_list', methods=['GET', 'POST'])
69 def update_course_list():
70     selected_city = json.loads(request.args.get('selected_city'))
71     selected_university = json.loads(request.args.get('selected_university'))
72     selected_course = json.loads(request.args.get('selected_course'))
73     universities = University.query.filter(University.university_name.in_(selected_university)).all()
74
75     university_ids = []
76     for uni in universities:
77         university_ids.append(uni.university_id)
78     html_string_selected = ''
79
80     if(selected_course):
81         course_ids = []
82         if (not selected_city and not selected_university):
83             course_ids = Course.query.filter(Course.course_name.in_(selected_course)).all()
84         elif (selected_university):
85             course_ids = Course.query.filter(Course.course_name.in_(selected_course), Course.university_id.in_(university_ids)).all()
86         elif (selected_city):
87             unis = University.query.filter(University.location.in_(selected_city)).all()
88             university_ids.clear()
89             for uni in unis:
90                 university_ids.append(uni.university_id)
91             course_ids = Course.query.filter(Course.course_name.in_(selected_course), Course.university_id.in_(university_ids)).all()
92         for course in course_ids:
93             html_string_selected += """<a href="/university/course/{}"
94             class="list-group-item list-group-item-action flex-column align-items-start active">
95             <div class="d-flex w-100 justify-content-between"><h5 class="mb-1">{}</h5><small>{}</small></div><p class="mb-1">
96             Tytuł: {}</p><small>Typ: {}</small></a>""".format(course.course_id, course.course_name,
97             University.query.filter_by(university_id=course.university_id).first().university_name, course.degree, course.cycle)
98         return jsonify(html_string_selected=html_string_selected)
99
100     if(selected_university):
101         course_ids = Course.query.filter(Course.university_id.in_(university_ids)).all()
102         for course in course_ids:
103             html_string_selected += """<a href="/university/course/{}" class="list-group-item list-group-item-action flex-column align-items-start active">
104             <div class="d-flex w-100 justify-content-between"><h5 class="mb-1">{}</h5><small>{}</small></div><p class="mb-1">
105             </small></div><p class="mb-1">Tytuł: {}</p><small>Typ: {}</small></a>
106             """.format(course.course_id,
107             course.course_name, University.query.filter_by(university_id=course.university_id).first().university_name, course.degree, course.cycle)
108         return jsonify(html_string_selected=html_string_selected)
109
110     if(selected_city):
111         unis = University.query.filter(University.location.in_(selected_city)).all()
112         university_ids.clear()
113         for uni in unis:
114             university_ids.append(uni.university_id)
115         course_ids = Course.query.filter(Course.university_id.in_(university_ids)).all()
116         for course in course_ids:
117             html_string_selected += """<a href="/university/course/{}"
118             class="list-group-item list-group-item-action flex-column align-items-start active">
119             <div class="d-flex w-100 justify-content-between"><h5 class="mb-1">{}</h5><small>{}</small></div><p class="mb-1">
120             Tytuł: {}</p><small>Typ: {}</small></a>""".format(course.course_id, course.course_name,
121             University.query.filter_by(university_id=course.university_id).first().university_name, course.degree, course.cycle)
122         return jsonify(html_string_selected=html_string_selected)
123
124     if (html_string_selected==''):
125         flash('Brak kursów!', category = 'error')
126
127     return jsonify(html_string_selected=html_string_selected)

```



```
75     src="https://code.jquery.com/jquery-3.6.0.min.js"
76 ></script>
77 <script src="https://cdn.jsdelivr.net/npm/select2@4.1.0-rc.0/dist/js/select2.min.js"></script>
78 <script type="text/javascript">
79     $(document).ready(function () {
80         $("#all_cities").select2();
81         $("#all_universities").select2();
82         $("#all_courses").select2();
83         $("#all_cities").on('change', function () {
84             jQuery.getJSON("{ url_for('views.update_university_dropdown') }", {
85                 selected_city: JSON.stringify($("#all_cities").select2("val")),
86             }).done(function (selected_city) {
87                 $("#all_universities").html(selected_city.html_string_selected);
88             }).fail(function(jqXHR, textStatus, errorThrown) {
89             });
90         });
91         $("#all_universities").on('change', function () {
92             jQuery.getJSON("{ url_for('views.update_course_dropdown') }", {
93                 selected_university: JSON.stringify($("#all_universities").select2("val")),
94             }).done(function (selected_university) {
95                 $("#all_courses").html(selected_university.html_string_selected);
96             }).fail(function(jqXHR, textStatus, errorThrown) {
97             });
98         });
99         $("#process_input").bind("click", function () {
100             jQuery.getJSON("{ url_for('views.update_course_list') }", {
101                 selected_city: JSON.stringify($("#all_cities").select2("val")),
102                 selected_university: JSON.stringify($("#all_universities").select2("val")),
103                 selected_course: JSON.stringify($("#all_courses").select2("val")),
104             }).done(function (selected_city) {
105                 $("#course_list").html(selected_city.html_string_selected);
```

# Schemat BAZY DANYCH



# TESTOWANIE APLIKACJI



pytest



**Pytest-Flask: to rozszerzenie dla biblioteki  
pytest, które ułatwia testowanie aplikacji  
opartych na Flasku. Zapewnia funkcje  
specyficzne dla Flaska, takie jak tworzenie  
klienta testowego, wstrzykiwanie aplikacji do  
testów. Dzięki temu można pisać testy  
jednostkowe i integracyjne w prosty sposób.**

**Pytest:** Pytest to platforma testowa Pythona, która wywodzi się z projektu PyPy. Może być używany do pisania różnego rodzaju testów oprogramowania, w tym testów jednostkowych, testów integracyjnych, testów kompleksowych i testów funkcjonalnych. Popularna ze względu na swoją prostotę, elastyczność i bogate funkcjonalności. Pytest łatwo integruje się z innymi narzędziami i bibliotekami, co umożliwia testowanie w kontekście innych technologii, takich jak framework Flask używany w naszym projekcie.

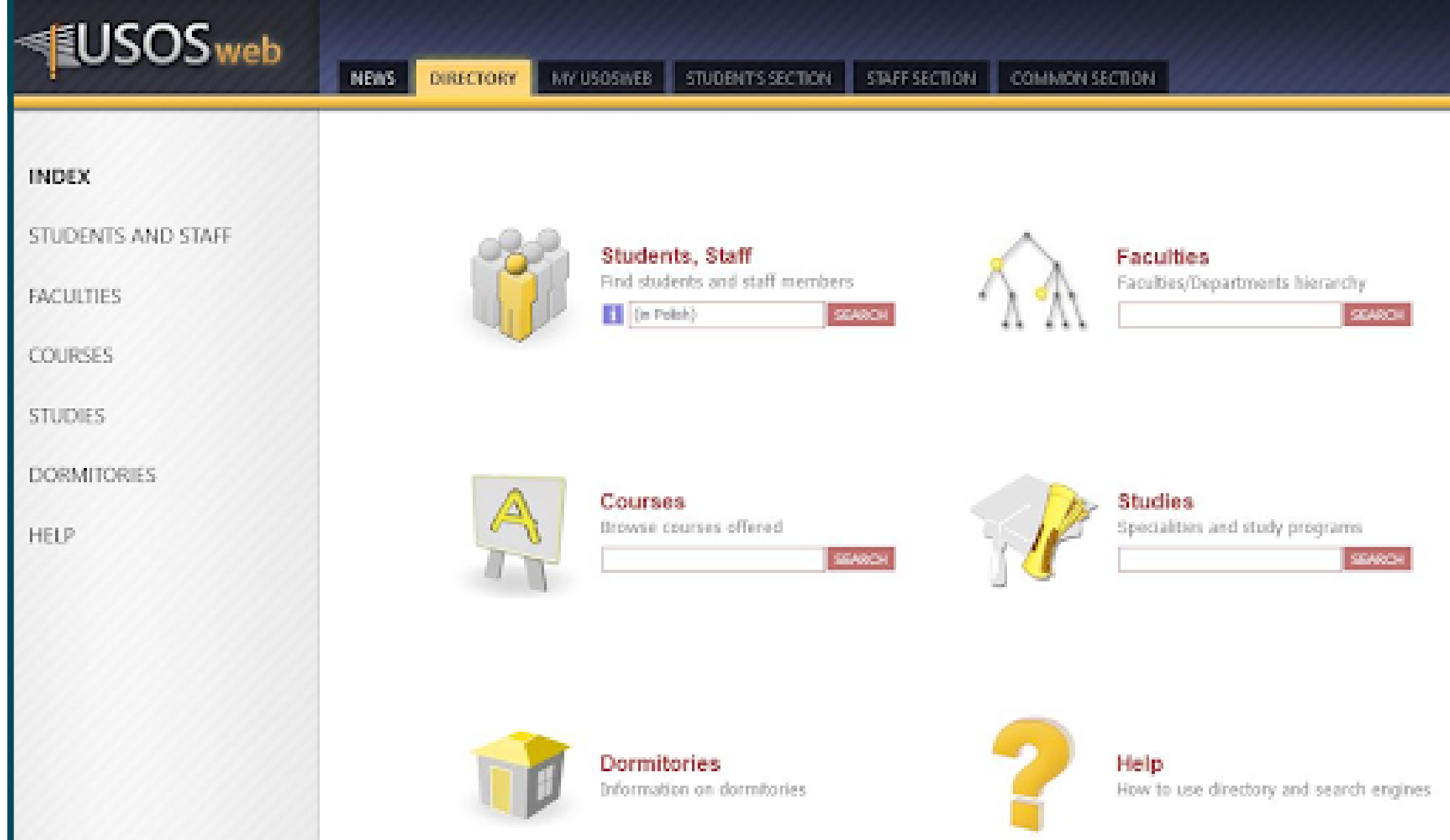


# BEZPIECZEŃSTWO



# BEZPIECZNE PRZETWARZANIE DANYCH

Witryna nie ma problemu z wyciekami danych ponieważ cała baza nie jest trzymana na naszym serwerze lecz dane są na bieżąco pobierane. Witryna uwierzytelnia opinie przekierowując użytkownika do strony logowania USOS. Witryna zapewnia zabezpieczenia przetwarzania danych w trakcie ich przesyłania oraz przechowuje, zgodnie z wymaganiami ogólnego rozporządzenia o ochronie danych (RODO).



**Dodatkowo jest zapewniona kontrola dostępu do różnych poziomów uprawnień użytkowników.**

# DOBRE PRAKTYKI PROGRAMOWANIA W JĘZYKU PYTHON

KTÓRE MIELIŚMY NA UWADZE TWORZĄC PROJEKT



# Co mieliśmy na uwadze:



**KISS - Keep It Stupid Simple**

**PEP (20) - Python Enhancement Proposals**

to dokument zawierający propozycję dotyczącą rozwoju języka Python. PEPy są tworzone przez społeczność Pythona i służą jako środek komunikacji i dokumentacji dla wprowadzenia nowych funkcji, zmian w istniejących funkcjach, standardów czy praktyk programistycznych.

**DRY - Don't Repeat Yourself**

Unikanie powtarzania kodu i dzielenie go na mniejsze funkcje i klasy, które wykonują konkretne zadania ułatwia. czytanie, testowanie i utrzymanie kodu

# NASZ ZESPÓŁ



# DONE

21

DONE

...

Dodac użytkowników z różnymi uprawnieniami

IO2-18

✓

=

DD

Rewizja procedury logowania.

IO2-26

✓

=

DD

jednolita stylizacja wszystkich stron (np. stworzenie style.css)

IO2-32

✓

=

NG

Prezentacja projektu

IO2-34

✓

=

NW

Dodanie scrolowania/stron do strony głównej

IO2-30

✓

=

DD

Strona konkretnego uniwersytetu (fullstack)

IO2-31

✓

=

NG

Stworzyć pola na opinie

IO2-12

✓

^

DD

Stworzyc poszczególne kierunki

IO2-13

✓

^

NG

Dodać wysyłanie powiadomień na maila

IO2-10

✓

=

IG

Stworzyc homepage(st.wizualna i techniczna)

IO2-7

✓

^

DD

Stworzyc strone logowania

IO2-9

✓

=

MW

Stworzyc baze danych

IO2-11

✓

=

NG

Stworzyc strone rejestracji

IO2-8

✓

=

V

Szyfrowanie hasła

IO2-24

✓

=

MW

Frontend for Homepage

IO2-21

✓

^

NG

Prezentacja na zajęcia

IO2-28

✓

=

NW

Wypelnic kierunki z bazy danych

IO2-16

✓

^

NG

Strona konkretnego kierunku (fullstack)

IO2-25

✓

=

MW

Ujednolicenie bazy danych

IO2-27

✓

=

NG

Stworzenie Logo

IO2-33

✓

=

IG

Zmiana struktury strony głównej (wyświetlanie uniwersytetów gdy nie są wybrane żadne kierunki)

IO2-29

✓

=

NG



# IN PROGRESS

IN PROGRESS 5

Dodać możliwość zadawania pytań

✓ IO2-35

= MW

Utworzenie dokumentacji

✓ IO2-23

= NW

Połączyć z baza USOS

✓ IO2-17

^ V

Stworzyć weryfikację użytkownika przez USOS

✓ IO2-15

≡ V

Stworzyć ranking uczelni

✓ IO2-14

∨ MW

# TO DO

⋮

TO DO 2

⋮

Testowanie aplikacji

☒

IO2-22

=

NW

Dodanie autentykacji admina

☒

IO2-36

=

V

+ Create

# CZEGO SIĘ NAUCZYLIŚMY?



**Git:** jest rozproszonym systemem kontroli wersji, który pozwala na skuteczne zarządzanie kodem źródłowym projektu. Pracując nad projektem zespołowym, można nauczyć się korzystać z podstawowych funkcji Gita, takich jak tworzenie repozytorium, tworzenie gałęzi (branching), wykonywanie commitów, scalanie zmian (merging) oraz rozwiązywanie konfliktów. Git umożliwia skuteczne śledzenie historii zmian w kodzie, zapewniając elastyczność i bezpieczeństwo podczas pracy nad projektem zespołowym.

# CZEGO SIĘ NAUCZYLIŚMY?



Podstawowa architektura aplikacji webowych: Projektowanie i implementacja aplikacji webowych wymaga zrozumienia podstawowych koncepcji związanych z architekturą webową. Przy tworzeniu projektu można zdobyć wiedzę na temat podziału aplikacji na warstwy (np. warstwa prezentacji, warstwa logiki biznesowej, warstwa danych), zarządzania danymi oraz zarządzania zasobami (np. pliki statyczne, cache). Poznanie tych podstawowych koncepcji pozwala na tworzenie skalowalnych, modułowych i efektywnych aplikacji webowych..

# CZEGO SIĘ NAUCZYLIŚMY?



Oprócz powyższych aspektów, przy realizacji projektu można również nauczyć się wielu innych umiejętności, takich jak planowanie i zarządzanie projektem, komunikacja i współpraca zespołowa, rozwiązywanie problemów technicznych, testowanie i debugowanie kodu, dostosowywanie się do zmieniających się wymagań oraz zdobywanie doświadczenia w pracy nad rzeczywistym projektem.



# OD ZERA DO PYTHON DEVELOPERA

