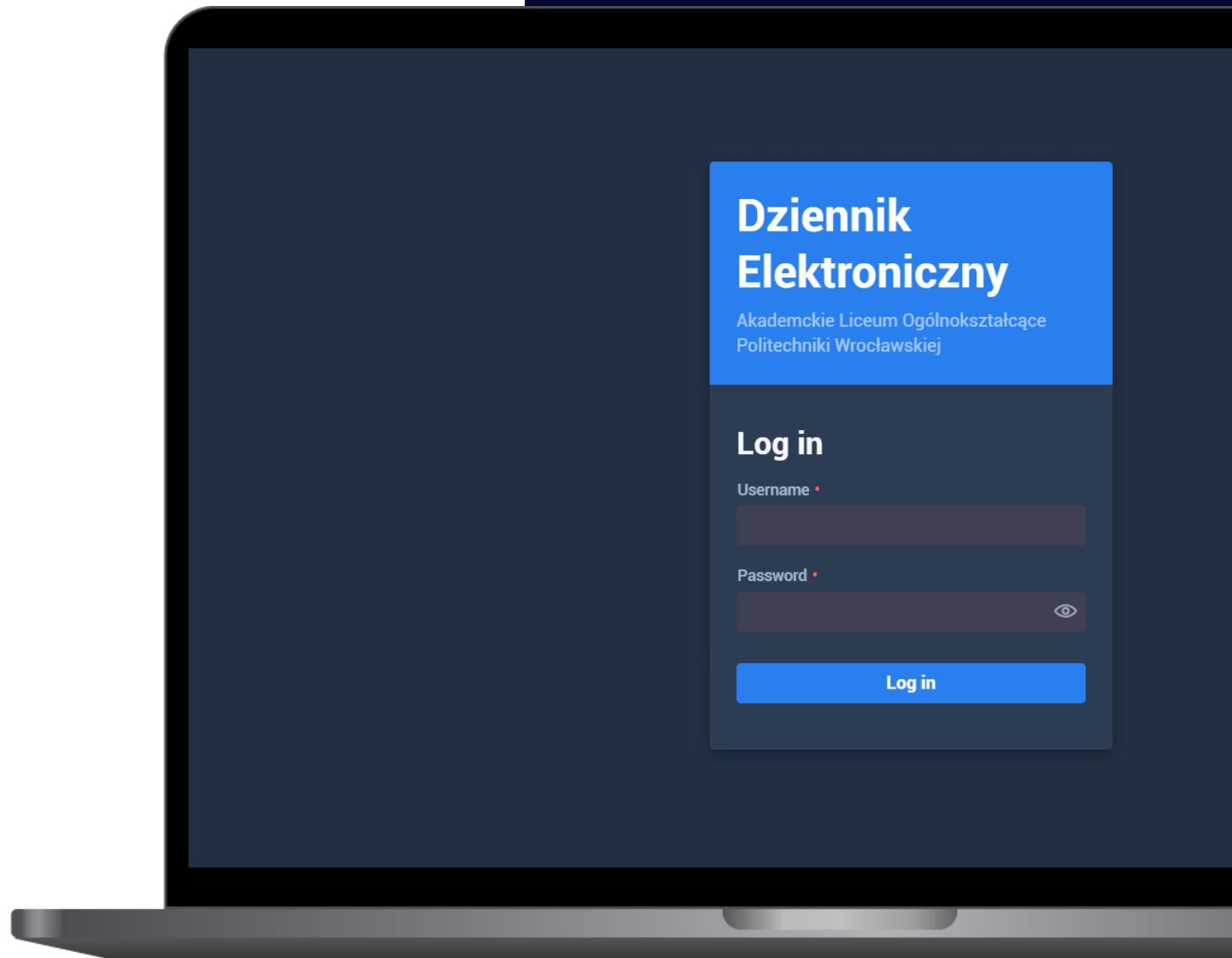
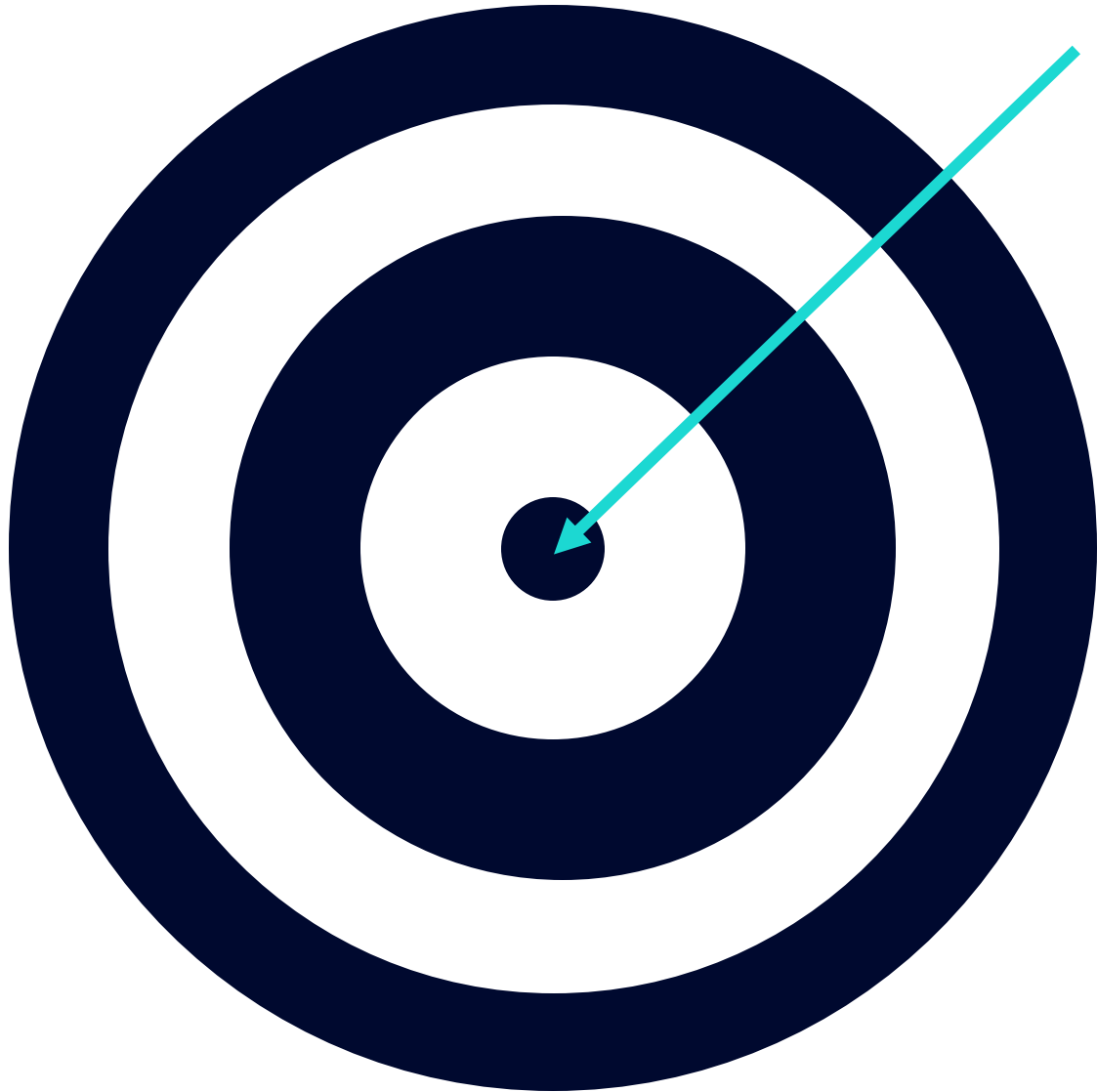


Dziennik elektroniczny

Mikołaj Chmielecki
Natalia Rusin
Aleksandra Tłałka





Cel:

Celem projektu było zaprojektowanie bazy służącej do obsługi **dziennika elektronicznego** w Akademickim Liceum Ogólnokształcącym Politechniki Wrocławskiej. Baza powinna nie tylko być skoncentrowana na **zarządzaniu ocenami, frekwencją, sprawdzianami lub lekcjami**, ale także dawać możliwość zarządzania klasami, uczniami czy tworzenia planu zajęć.

Wykorzystane narzędzia



Selenium IDE 3.17



Java 16



IntelliJ 2021.3



Hibernate 5.6



Vaadin 14.8



Spring 2.6



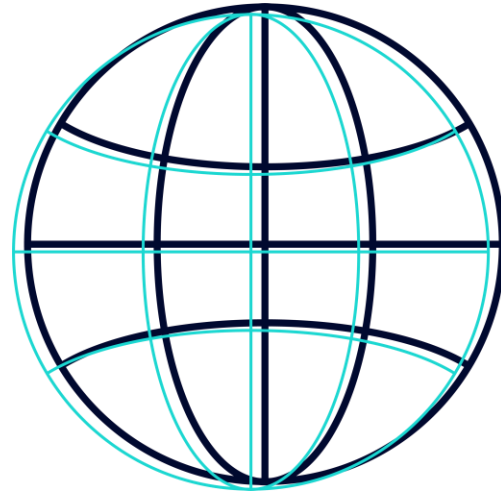
Visual Paradigm

Visual Paradigm 16.3



MySQL 8.0

Wymagania i założenia



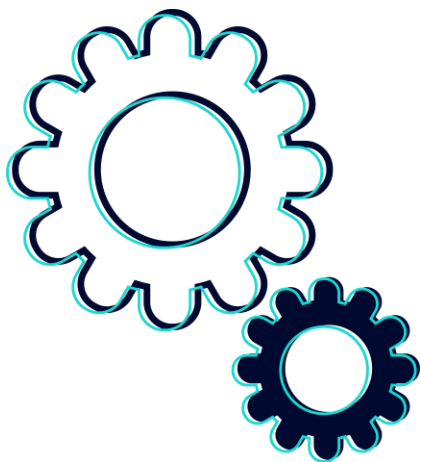
Dziennik Elektroniczny jest aplikacją webową

Każdy użytkownik posiada konto

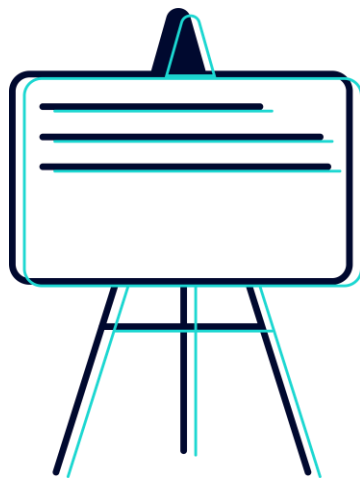
Brak redundancji danych

Konto rodzica obsługuje wszystkie jego dzieci

Archiwizacja przez okres dziesięciu lat



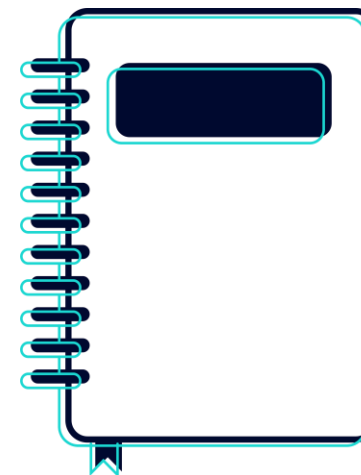
Administrator



Nauczyciel



Rodzic



Uczeń

Użytkownicy

Administrator

1

- Dodawanie, usuwanie, edycja klas
- Archiwizowanie klas, zajęć i użytkowników
- Tworzenie i zmiana planów zajęć
- Przypisywanie nauczycieli do zajęć
- Dodawanie użytkowników
- Edycja danych użytkowników

Nauczyciel

2

- Wpisywanie, zmiana i usuwanie tematów lekcji
- Zaznaczanie i odznaczanie nieobecności uczniów na zajęciach
- Wpisywanie, edycja i usuwanie ocen swoim uczniom
- Wgląd w swój plan zajęć oraz w plan lekcji klasy
- Zapowiadanie, zmiana terminu, odwoływanie i przeglądanie zapowiedzianych przez siebie sprawdzianów
- Usprawiedliwianie nieobecności uczniów

Rodzic

3

- Przeglądanie ocen dziecka
- Sprawdzanie planu zajęć dziecka
- Przeglądanie zapowiedzianych dla niego sprawdzianów
- Sprawdzenie frekwencji swoich dzieci i usprawiedliwianie ich nieobecności

Uczeń

4

- Przeglądanie własnych ocen
- Sprawdzanie swojego planu zajęć
- Przeglądanie zapowiedzianych dla niego sprawdzianów
- Sprawdzenie swojej frekwencji

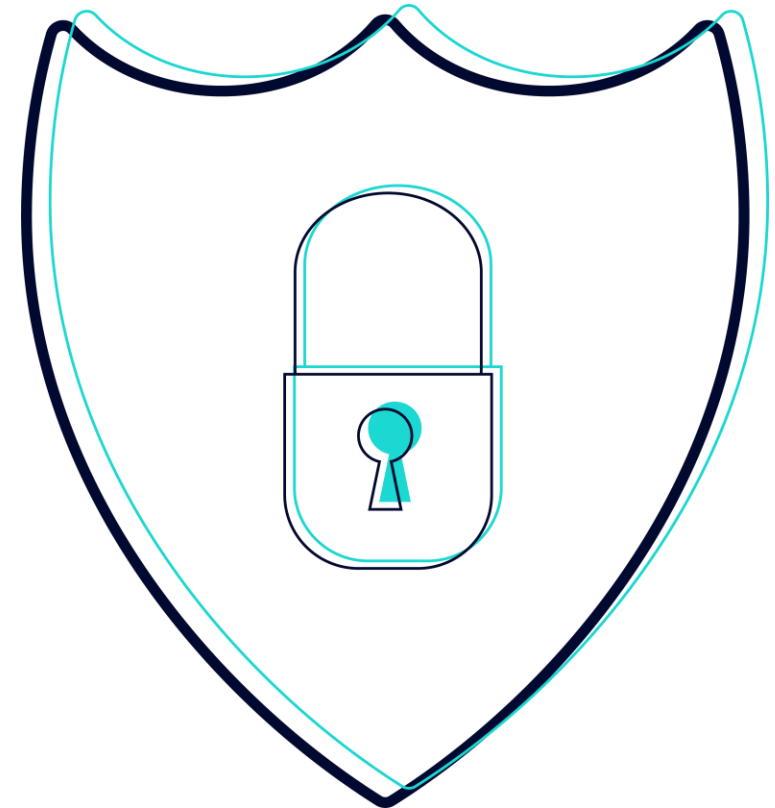
Bezpieczeństwo

Serwerownia ma odpowiednie warunki

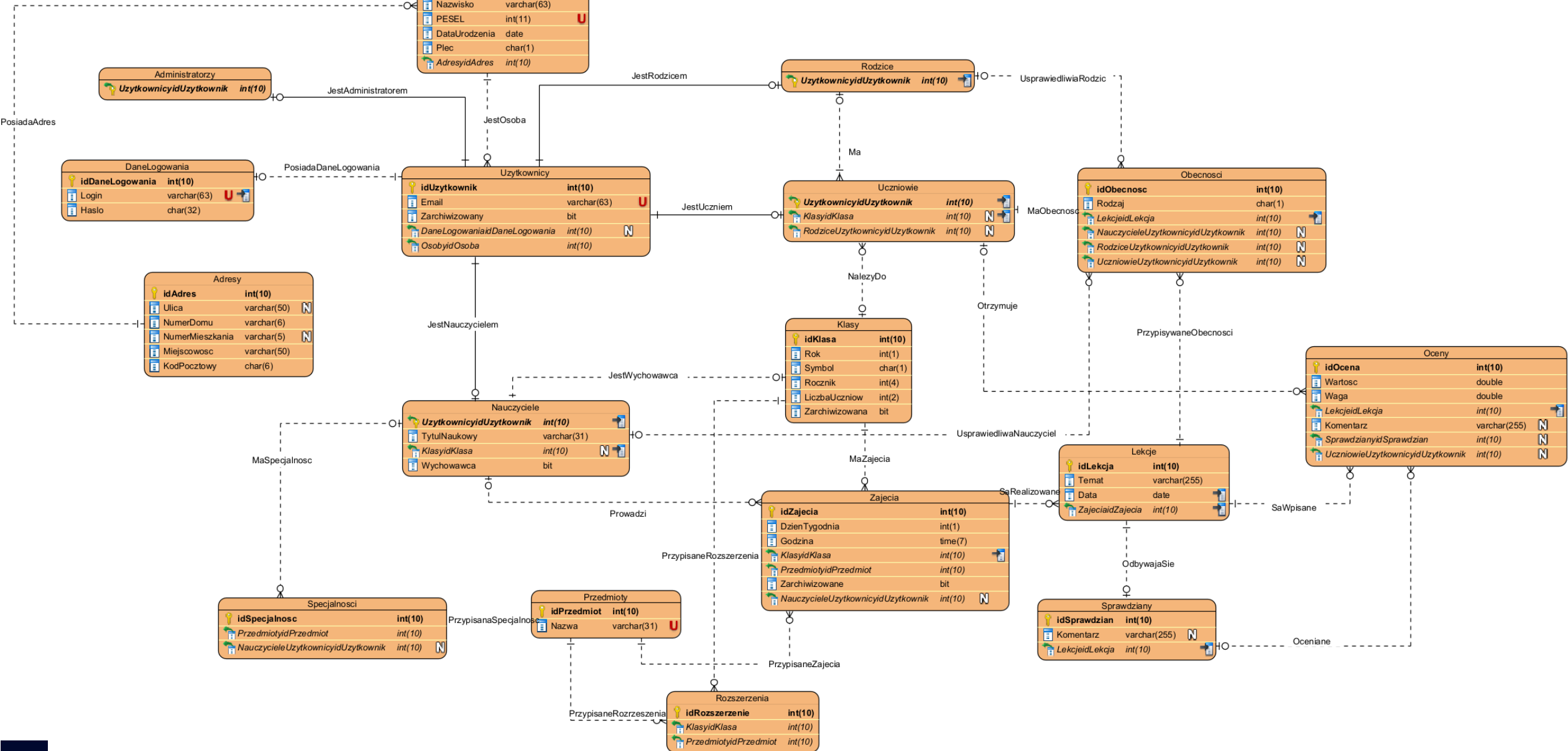
Aplikacja dostępna publicznie

Chroniony dostęp do bazy

Użytkownicy mają login i hasło do swoich kont



Model bazy danych



Projekt bazy danych

Widoki

Procedury Składowane

Indeksy

Triggery

Granty

Przykładowa implementacja bazy danych







```
CREATE TABLE Uczniowie (  
  idUzytkownik int(10) NOT NULL,  
  KlasyidKlasa int(10),  
  RodziceidRodzic int(10),  
  PRIMARY KEY (idUzytkownik)) CHARACTER SET UTF8;  
  
ALTER TABLE Uczniowie ADD CONSTRAINT Ma  
FOREIGN KEY (RodziceidRodzic) REFERENCES Rodzice (idUzytkownik);  
  
ALTER TABLE Uczniowie ADD CONSTRAINT NalezyDo  
FOREIGN KEY (KlasyidKlasa) REFERENCES Klasy (idKlasa);  
  
DROP procedure IF EXISTS `wprowadzOcene`;  
DELIMITER $$  
USE `dziennikszkolny`$$  
CREATE PROCEDURE `wprowadzOcene`  
(ocena integer, waga double, idLekcja integer, idUczen integer, idSprawdzian integer, komentarz varchar(255))  
BEGIN  
  INSERT INTO Oceny (Wartosc, Waga, UczniowieidUczen, LekcjeidLekcja, SprawdzianyidSprawdzian, Komentarz)  
  values (ocena, waga, idLekcja, idUczen, idSprawdzian, komentarz);  
END$$  
DELIMITER ;
```

```
CREATE INDEX LekcjeDataDescIdx ON Lekcje (Data DESC);  
CREATE INDEX OsobyIdx ON Osoby (Imie ASC, DrugieImie ASC, Nazwisko ASC);  
  
DELIMITER /  
CREATE TRIGGER LimitUczniowInsert  
BEFORE INSERT ON Uczniowie  
FOR EACH ROW  
BEGIN  
  DECLARE  
    liczba INTEGER;  
  SELECT COUNT(*) INTO liczba FROM Uczniowie  
  WHERE Uczniowie.KlasyidKlasa = NEW.KlasyidKlasa;  
  
  IF (liczba>30) THEN  
    signal sqlstate '45000' set message_text = 'Klasa jest pelna';  
  END IF;  
END;/  
DELIMITER ;
```

Testowanie bazy danych

SELECT * FROM dziennikszkolny.oceny;

1 • SELECT * FROM dziennikszkolny.oceny;

<							
Result Grid							
Filter Rows: <input type="text"/>							
Edit:   							
Export/Import:  							
Wrap Cell Content: 							
	idOcena	UcniowieidUczen	LekcjeidLekcja	Wartosc	Waga	Komentarz	SprawdzianyidSprawdzian
▶	1	1	3	5	3	Nunc ac blandit augue.	1
	2	2	3	4	3	Nunc ac blandit augue.	1
	3	3	3	5	3	Nunc ac blandit augue.	1
	4	4	3	3	3	Nunc ac blandit augue.	1
	5	5	3	5	3	Nunc ac blandit augue.	1
	6	6	3	6	3	Nunc ac blandit augue.	1
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Wstawianie danych korzystając z gotowych procedur składowanych

Call stored procedure dziennikszkolny.wprowadzOcene

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

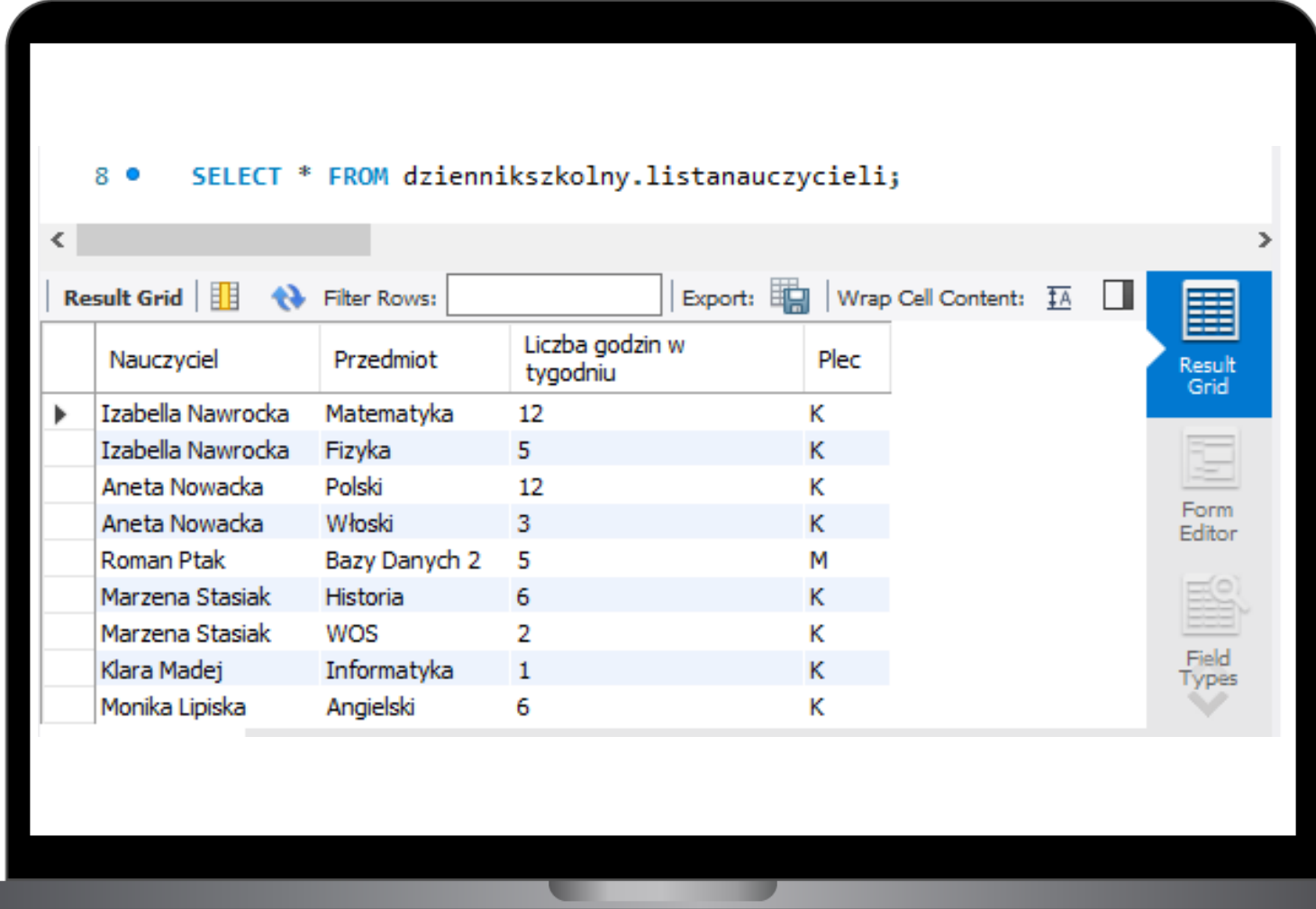
ocena	<input type="text" value="3"/>	[IN] integer
waga	<input type="text" value="2"/>	[IN] double
idLekcja	<input type="text" value="5"/>	[IN] integer
idUczen	<input type="text" value="3"/>	[IN] integer
idSprawdzian	<input type="text"/>	[IN] integer
komentarz	<input type="text" value="Ocena za odpowiedz"/>	[IN] varchar(255)

```
1 CREATE DEFINER='root'@'localhost' PROCEDURE `wprowadzOcene`  
2 (ocena integer, waga double, idLekcja integer,  
3 idUczen integer, idSprawdzian integer,  
4 komentarz varchar(255))  
5 BEGIN  
6 INSERT INTO Oceny (Wartosc, Waga, UczniowieidUczen,  
7 LekcjeidLekcja, SprawdzianyidSprawdzian, Komentarz)  
8 values (ocena, waga, idLekcja, idUczen,  
9 idSprawdzian, komentarz);  
10 END
```

```
SELECT DISTINCT TABLE_NAME, INDEX_NAME
FROM INFORMATION_SCHEMA.STATISTICS
WHERE TABLE_SCHEMA = `DziennikSzkolny`;
```

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	NON_UNIQUE	INDEX_SCHEMA	INDEX_NAME	SEQ_IN_INDEX	COLUMN_NAME
def	dziennikszkolny	przedmioty	0	dziennikszkolny	Nazwa	1	Nazwa
def	dziennikszkolny	adresy	0	dziennikszkolny	PRIMARY	1	idAdres
def	dziennikszkolny	specjalnosci	0	dziennikszkolny	PRIMARY	1	idSpecjalnosc
def	dziennikszkolny	specjalnosci	1	dziennikszkolny	PrzypisanaSpecjalnosc	1	PrzedmiotyidPrzedmiot
def	dziennikszkolny	specjalnosci	1	dziennikszkolny	MaSpecjalnosc	1	NauczycieleidNauczyciel
def	dziennikszkolny	rodzice	0	dziennikszkolny	PRIMARY	1	idUzytkownik
def	dziennikszkolny	administratorzy	0	dziennikszkolny	PRIMARY	1	idUzytkownik
def	dziennikszkolny	nauczyciele	0	dziennikszkolny	PRIMARY	1	idUzytkownik
def	dziennikszkolny	rozszerzenia	0	dziennikszkolny	PRIMARY	1	idRozszerzenie
def	dziennikszkolny	rozszerzenia	1	dziennikszkolny	PrzypisaneRozszerzenia	1	KlasyidKlasa
def	dziennikszkolny	rozszerzenia	1	dziennikszkolny	PrzypisaneRozszerzenia	1	PrzedmiotyidPrzedmiot
def	dziennikszkolny	osoby	0	dziennikszkolny	PRIMARY	1	idOsoba
def	dziennikszkolny	osoby	0	dziennikszkolny	PESEL	1	PESEL
def	dziennikszkolny	osoby	1	dziennikszkolny	PosiadaAdres	1	AdresyidAdres
def	dziennikszkolny	osoby	1	dziennikszkolny	OsobyIdx	1	Imie
def	dziennikszkolny	osoby	1	dziennikszkolny	OsobyIdx	2	DrugieImie
def	dziennikszkolny	osoby	1	dziennikszkolny	OsobyIdx	3	Nazwisko
def	dziennikszkolny	uzytkownicy	0	dziennikszkolny	PRIMARY	1	idUzytkownik
def	dziennikszkolny	uzytkownicy	0	dziennikszkolny	Email	1	Email
def	dziennikszkolny	uzytkownicy	1	dziennikszkolny	JestOsoba	1	OsobyidOsoba

SELECT * FROM
dziennikszkolny.listanauczycieli; - widok




8 • SELECT * FROM dziennikszkolny.listanauczycieli;

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Nauczyciel	Przedmiot	Liczba godzin w tygodniu	Plec
▶	Izabella Nawrocka	Matematyka	12	K
	Izabella Nawrocka	Fizyka	5	K
	Aneta Nowacka	Polski	12	K
	Aneta Nowacka	Włoski	3	K
	Roman Ptak	Bazy Danych 2	5	M
	Marzena Stasiak	Historia	6	K
	Marzena Stasiak	WOS	2	K
	Klara Madej	Informatyka	1	K
	Monika Lipiska	Angielski	6	K

Result Grid
Form Editor
Field Types

Projekt Aplikacji



DZIENNIK ELEKTRONICZNY

Akademckie Liceum Ogólnokształcące
Politechniki Wrocławskiej

LOGIN

HASŁO

ZALOGUJ SIĘ

Witaj
Imię Nazwisko!

KLASA: 3A
MATEMATYKA X FIZYKA X INFROMATYKA

PLAN ZAJĘĆ

Godzina	Poniedziałek	Wtorek	Środa	Czwartek	Piątek
hh:mm	Matematyka Imię i Nazwisko			Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko
hh:mm	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko
hh:mm	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko
hh:mm	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko
hh:mm	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko
hh:mm	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko	Matematyka Imię i Nazwisko
hh:mm	Matematyka Imię i Nazwisko		Matematyka Imię i Nazwisko		

Diagram przypadków użycia

Frontend Aplikacji



Combo Box

Klasa

▼

2A

3B

Okno dialogowe

Dodaj ocenę

Wybierz ucznia

▼

Wybierz lekcję

▼

Wartość oceny

Waga oceny

Komentarz do oceny

Anuluj Zapisz

Szuflada

Witaj

Dorota Sikorowska

2A

Rozszerzenia:
Matematyka, Fizyka,
Informatyka,

Oceny

Przycisk

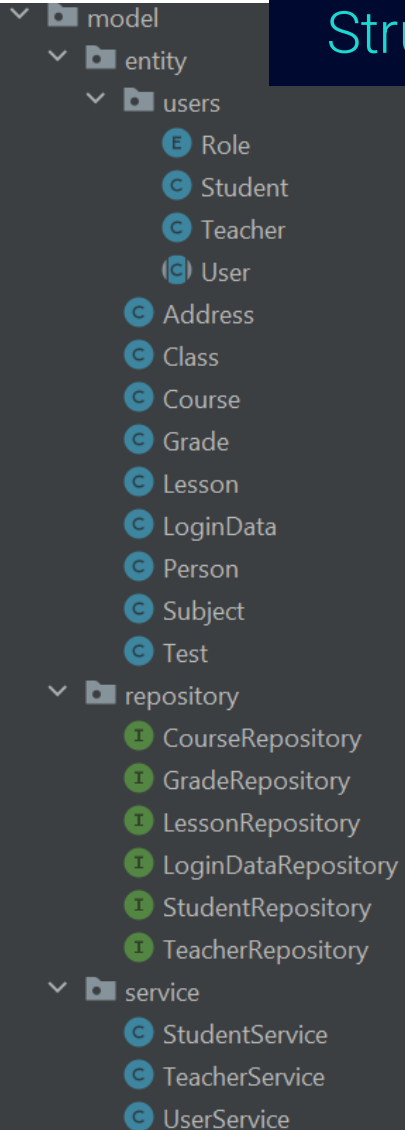
Anuluj Zapisz

Grid

Imię	Nazwisko	Ocena	Waga	Data	Komentarz		
Lucyna	Tomczyk	3.0	2.0	2021-11-08	Komentarz, który ma dwie linijki	Modyfikuj	Usuń
Fabian	Kowalczyk	6.0	3.0	2021-11-08	Test	Modyfikuj	Usuń

Backend aplikacji

Struktura projektu



```
@Repository
public interface StudentRepository extends JpaRepository<Student, Long> {
    @Query("select s from Student s where s.loginData.login = ?1")
    Student findByUsername(String username);

    @Query("select s from Student s where s.studentClass.id = ?1")
    List<Student> findByClassId(Long idClass);
}
```

Repozytorium - '@Repository'

```
@Entity
@Table(name = "uzytkownicy")
@Inheritance(strategy = InheritanceType.JOINED)
public abstract class User implements UserDetails {

    @Id
    @GeneratedValue
    @Column(name = "idUzytkownik")
    private Long id;
```

Dziedziczenie encji

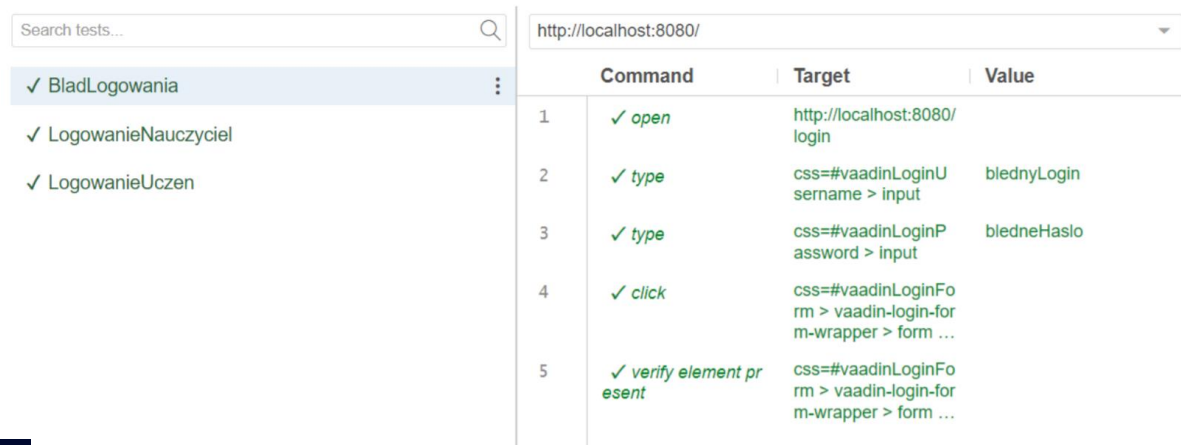
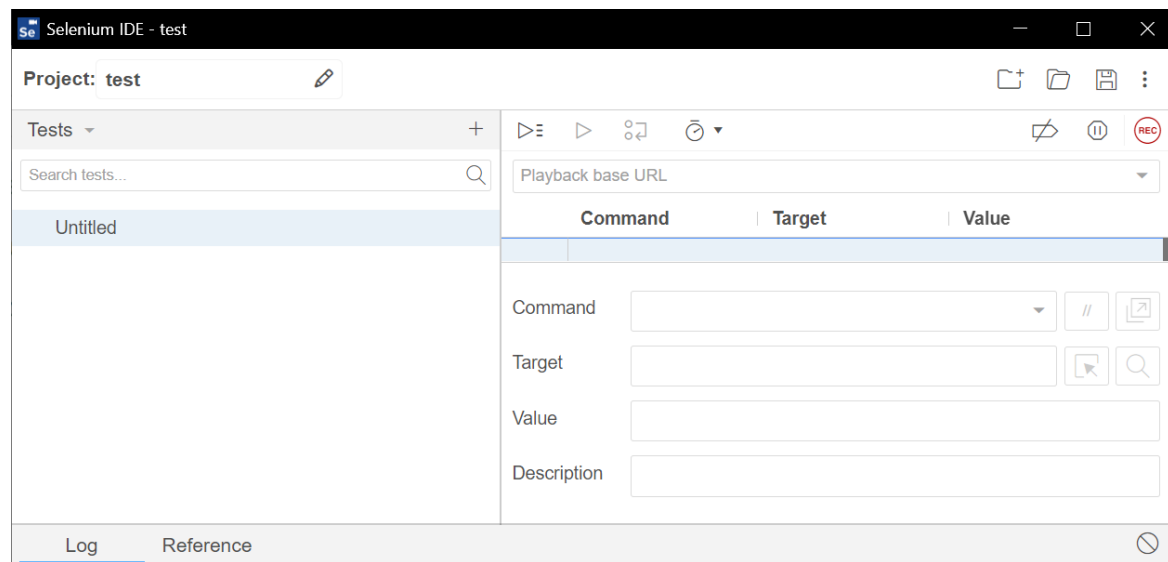
```
@Entity
@Table(name = "Uczniowie")
public class Student extends User {
```

Encja - '@Entity'

```
11 @Service
12 public class StudentService {
13
14     @Autowired
15     private StudentRepository studentRepository;
16
17     public List<Student> list() { return studentRepository.findAll(); }
```

Serwis - '@Service'

Testy Aplikacji



Test logowania ucznia

Test logowania nauczyciela

Test błędnego logowania

Podsumowanie



- Pełna funkcjonalność związana z ocenami dla ucznia

- Pełna funkcjonalność związana z ocenami dla nauczyciela

- Intuicyjny interfejs

- Poprawny proces logowania i autoryzacji użytkowników



- Utrwalenie i poszerzenie wiedzy z zakresu modelowania baz danych

- Stworzenie aplikacji webowej w oparciu o założenia i mockupy

- Nabycie wiedzy z zakresu tworzenia dokumentacji technicznej

- Spring, Hibernate, Vaadin

Źródła - dokumentacje



Dokumentacja Spring Boot: <https://docs.spring.io/spring-boot/docs/current/api/>



Dokumentacja Spring Security: <https://docs.spring.io/spring-security/site/docs/current/api/>



Dokumentacja Hibernate: <https://hibernate.org/orm/documentation/5.6/>



Dokumentacja Vaadin: <https://vaadin.com/docs/v14/>

Źródła - tutoriala



Tutorial Vaadin: <https://www.youtube.com/watch?v=bxy2JgqqKDU>



Tutorial Spring: <https://www.youtube.com/watch?v=9SGDpanrc8U>



Tutorial dziedziczenia w Hibernate: <https://thorben-janssen.com/complete-guide-inheritance-strategies-jpa-hibernate/>



Tutorial autoryzacji użytkowników w Spring Security:
<https://www.youtube.com/watch?v=oMKks5AjaSQ>

DZIENNIK ELEKTRONICZNY

Nowoczesna szkoła na Twoim komputerze

DZIENNIK ELEKTRONICZNY

Akademickie Liceum Ogólnokształcące
Politechniki Wrocławskiej

LOGIN

HASŁO

ZALOGUJ SIĘ

KWENCJA

WIEDZECNOŚCI

Szczegóły

● sprawdził ● nie sprawdzil

DD-MM-RR

nieobecności: 3 w tym: 0 sprawdził 0 nie sprawdzil

Nazwa przedmiotu	Godzina	Nauczyciel	Status
Matematyka	7:30-8:15	Imię i Nazwisko	●
Matematyka	8:25-9:10	Imię i Nazwisko	●
Polski	9:20-10:05	Imię i Nazwisko	●

DD-MM-RR

nieobecności: 5 w tym: 0 sprawdził 0 nie sprawdzil

Nazwa przedmiotu	Godzina	Nauczyciel	Status
Matematyka	7:30-8:15	Imię i Nazwisko	●
Matematyka	8:25-9:10	Imię i Nazwisko	●
Polski	9:20-10:05	Imię i Nazwisko	●
Angielski	9:20-10:05	Imię i Nazwisko	●
WF	9:20-10:05	Imię i Nazwisko	●