

# **PAŃSTWOWA WYŻSZA SZKOŁA ZAWODOWA W NOWYM SĄCZU**

## **INSTYTUT TECHNICZNY**

### **BAZY DANYCH**

#### **PRACA PROJEKTOWA**

#### **Dziennik elektroniczny**

**Autorzy:** Piotr Bożek, Paweł Cetnarowski

**Grupa:** P1

**Kierunek:** Informatyka Stosowana 2IS(s)

**Prowadzący:** mgr inż. Nikodem Bulanda

**Nowy Sącz 2022**

## Spis treści

Cel projektu .....	4
Analiza .....	4
Wymagania .....	5
Wymagania funkcjonalne .....	5
Wymagania nauczyciela .....	5
Wymagania rodzica .....	5
Wymagania ucznia .....	6
Wymagania administratora .....	6
Wymaganie nie funkcjonalne .....	6
Diagram Przypadków użycia .....	7
Admistrador .....	7
Nauczyciel .....	8
Rodzic .....	8
Uczeń .....	9
Diagram ERD .....	9
Scenariusze przypadków użycia .....	10
Scenariusz nr: 1 .....	10
Scenariusz nr: 2 .....	10
Scenariusz nr: 3 .....	11
Scenariusz nr: 4 .....	11
Scenariusz nr: 5 .....	12
Scenariusz nr: 6 .....	13
Scenariusz nr: 7 .....	13
Scenariusz nr: 8 .....	14
Scenariusz nr: 9 .....	14
Scenariusz nr: 10 .....	15
Scenariusz nr: 11 .....	15
Scenariusz nr: 12 .....	16
Scenariusz nr: 13 .....	16
Scenariusz nr: 14 .....	16
Scenariusz nr: 15 .....	17
Scenariusz nr: 16 .....	17
Scenariusz nr: 17 .....	18
Scenariusz nr: 18 .....	18
Scenariusz nr: 19 .....	19

Scenariusz nr: 20 .....	19
Scenariusz nr: 21 .....	20
Scenariusz nr: 22 .....	20
Scenariusz nr: 23 .....	21
Scenariusz nr: 24 .....	21
Scenariusz nr: 25 .....	21
Scenariusz nr: 26 .....	22
Scenariusz nr: 27 .....	22
Opis technologii.....	23
Implementacja bazy danych – migracje .....	25
Komunikacja z bazą danych mysql .....	31
Harmonogram Pracy .....	34
Lista zadań oraz estymacja czasowa .....	35
Bilans .....	37
Wnioski .....	38
Link do repozytorium .....	39
Przykładowe zrzuty ekranu z aplikacji .....	39

## Cel projektu

Celem projektu jest stworzenie aplikacji bazodanowej do obsługi dziennika elektronicznego w taki sposób aby umożliwiał nadzór ucznia . Aplikacja będzie zarządzana z poziomu witryny internetowej. Rodzice oraz nauczyciele będą mieli dostęp do informacji związanych z uczniem. Administrator będzie mógł dodawać i usuwać uczniów oraz rodziców z listy uczniów oraz rodziców będzie mógł aktualizować dane osobowe oraz będzie miał dostęp do danych bazy

## Analiza

Po przeprowadzeniu wywiadu z nauczycielem który korzysta z dziennika elektronicznego i zadaniu mu pytań na temat pracy z taką aplikacją określił jasno że każdy nauczyciel ma swoje konto na które może się zalogować i pełnić dalsze funkcje ,konto zostaje założone po przez głównego administratora aplikacji. Po zalogowaniu nauczyciel może wpisywać oceny w przedmiotach które uczy dla swoich uczniów ponadto ma możliwość sprawdzania frekwencji dzięki której dostarcza informacje ważne do systemu związane z frekwencją ucznia ,gdzie frekwencja jest stosunkiem wszystkich godzin obecności ucznia do wszystkich godzin nie obecności ucznia dzięki czemu nauczyciel może sprawdzić procent obecności ucznia. Po zadaniu pytania dla nauczyciela jak wpisywane są ogłoszenia do ucznia nauczyciel powiedział , że ma taką możliwość po przez rubrykę ogłoszenia. Nauczyciel prowadzący lekcje jest zobowiązany do wpisywania tematu na każdej lekcji na podstawie sylabusu i zobowiązany jest do terminowych realizacji tematów według norm określonych w sylabusie czyli do danego zagadnienia ma daną liczbę godzin na realizację danego zagadnienia poprawność realizacji zagadnień z sylabusu może weryfikować administrator ,który ma obowiązek zgłoszenia dyrektorowi zaistniałych nieprawidłowości. Nauczyciel ma także dostęp do ocen do ocen ucznia w celu monitorowania średniej ucznia ocen pozytywnych oraz negatywnych. Każdy z nauczycieli ma dostęp do swojego planu zajęć który ustala administrator i zamieszcza go na stronie. Po zadaniu pytania dla nauczyciela co w razie pomyłki przy wprowadzaniu danych nauczyciel odpowiedział że ma możliwość modyfikowania takich informacji jak oceny czy tematy w razie gdyby system się zamknął albo nauczyciel popełnił błąd przy wprowadzaniu danych.

Po rozmowie z rodzicami rodzice jednoznacznie określili swoje wymogi w stosunku do aplikacji w szczególności ważną informacją jest to aby interfejs logowania i obsługa aplikacji była prosta następnymi wymaganiami jakie opisali rodzice to dostęp do frekwencji swoich dzieci oraz ocen w celu weryfikacji czy ich dzieci sprawują się prawidłowo ,ważną informacją dla rodziców jest również dostęp do ogłoszeń które wypisują nauczyciele .Rodzice także podkreślili że chcieli by mieć dostęp do planu lekcji swoich dzieci gdyby zaszła potrzeba odbioru lub zwolnienia dziecka ze szkoły .Niektórzy z rodziców wyrazili także potrzebę wglądu do tematów zajęć do ich realizacji dzięki czemu sami będą mogli weryfikować poprawne nauczanie nauczyciela w szkole

Uczniowie oczywiście będą mieli po zalogowaniu na swoje kąta wgląd do swoich ocen i frekwencji dostęp będą mieli także dostęp będą mieli także do swoich ogłoszeń oraz tematów lekcyjnych. Jednak najważniejszą funkcją dla uczniów jest dostęp szybki dostęp do planu lekcji aby mogli terminowo docierać na zajęcia .

Po rozmowie z administracją zostały stwierdzone obowiązki tej grup w stosunku do nauczycieli, rodziców i uczniów. Administracja będzie odpowiedzialna za tworzenie kąt dla nauczycieli, rodziców i uczniów oraz ich usuwanie w razie gdy uczeń zrezygnuje z nauczania w danej szkole .Administratorzy będą mogli także przypisywać nauczycieli do danego przedmiotu lub ich zmieniać w wypadku rezygnacji nauczyciela z nauczania lub innych przyczyn , w razie zmiany sylabusu będą

mieli możliwość dodania lub usunięcia przedmiotu .Administratorzy zwrócili uwagę także na możliwości modyfikacji ocen czy frekwencji uczniów w razie awaryjnej potrzeby gdyby nauczyciel zachorował lub sytuacja wiązała się z dłużą nieobecnością nauczyciela. Administratorzy będą odpowiedzialni za tworzenie także planów zajęć oraz o dbanie i funkcjonowanie aplikacji internetowej , jednym z obowiązków administratorów będzie także awarii aplikacji po przez umiejętne użytkowanie aplikacji

Po podsumowaniu oczekiwań czterech grup społecznych określiliśmy wymagania funkcjonalne i nie funkcjonalne które zapisaliśmy poniżej .

## Wymagania

### Wymagania funkcjonalne

#### Wymagania nauczyciela

Nauczyciel po zalogowaniu na swoje konto będzie mógł :

- Wpisywać oceny uczynią
- Zaznaczać obecność lub nie obecność
- Wpisywać ogłoszenia
- Wpisywać tematy wraz z liczbą przewidzianych
- Analizować statystyki ucznia takie jak oceny czy frekwencja (frekwencja jest stosunkiem godzin obecnych do nie obecnych)
- Przeglądać plany lekcji

#### Wymagania rodzica

Rodzic po zalogowaniu na swoje konto będzie mógł :

- Sprawdząć frekwencję dziecka (frekwencja jest stosunkiem godzin obecnych do nie obecnych)
- Wyświetlać ogłoszenia swojego dziecka
- Przeglądać oceny dziecka
- Przeglądać plany lekcji

### Wymagania ucznia

Uczeń po zalogowaniu na swoje konto będzie mógł :

- Sprawdzić swoją frekwencję  
(frekwencja jest stosunkiem godzin obecnych do nie obecnych)
- Wyświetlać swój plan
- Wyświetlić swoje oceny
- Podglądać zrealizowane tematy

### Wymagania administratora

Administrator po zalogowaniu na swoje konto będzie mógł :

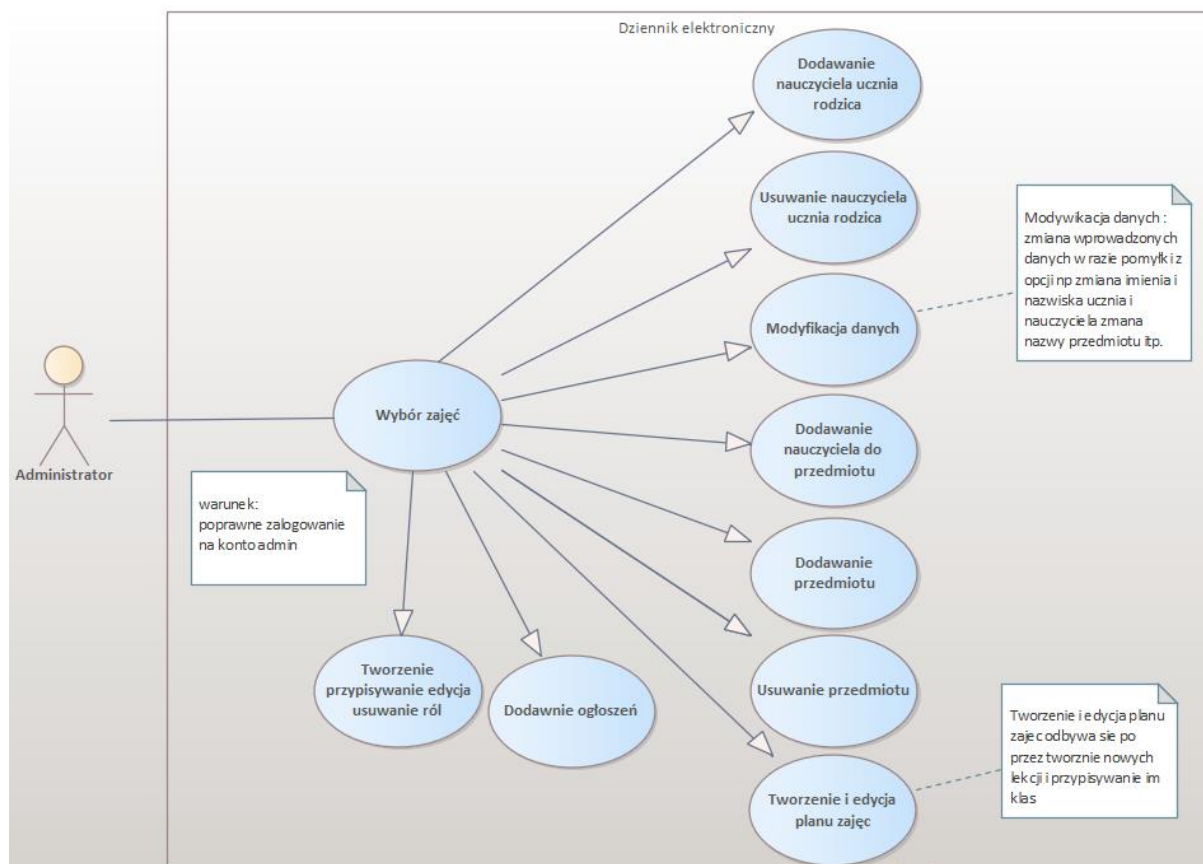
- Dodawać lub usuwać uczniów wraz z rodzicami oraz nauczycielami
- Dodawać lub usuwać przedmiot
- Tworzyć, edytować i usuwać role do których są przypisani użytkownicy
- Modyfikować uprawnienia dla danych ról
- Tworzyć lekcje, z których generowany będzie plan lekcji dla poszczególnych klas
- Modyfikować takie dane jak frekwencja czy oceny
- Dodawać nauczyciela do przedmiotu lub go zmieniać

### Wymaganie nie funkcjonalne

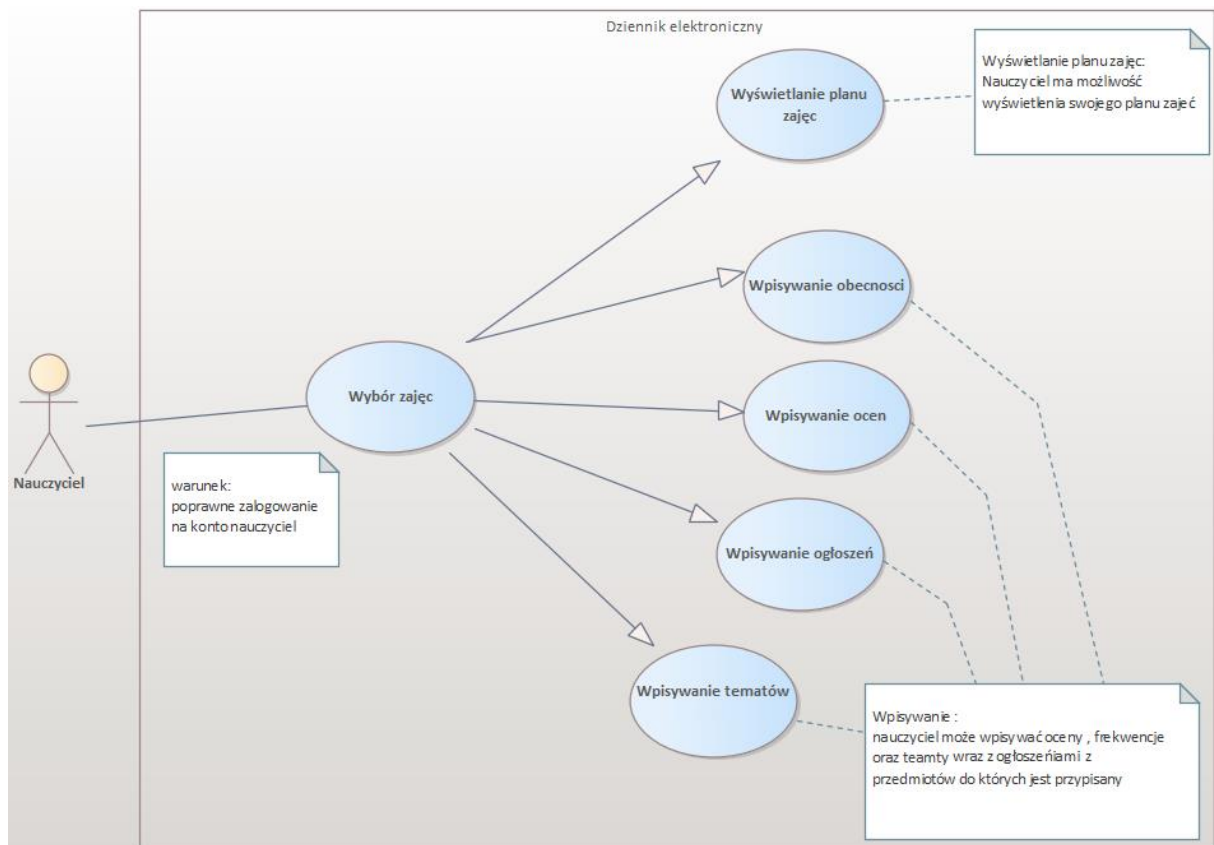
Na wymagania nie funkcjonalne będą się składać wygląd strony czy oprawa graficzna a także poufność i stopień zabezpieczeń

## Diagram Przypadków użycia

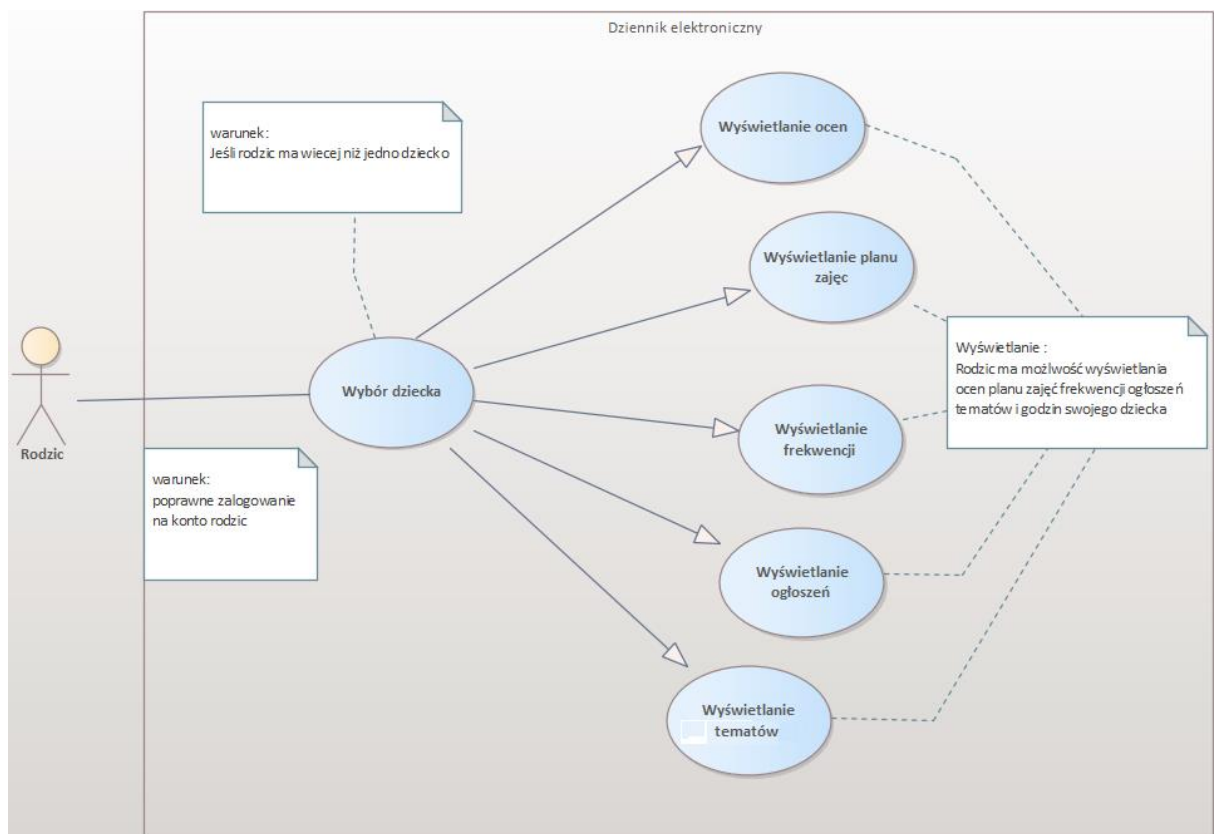
### Administrador



## Nauczyciel

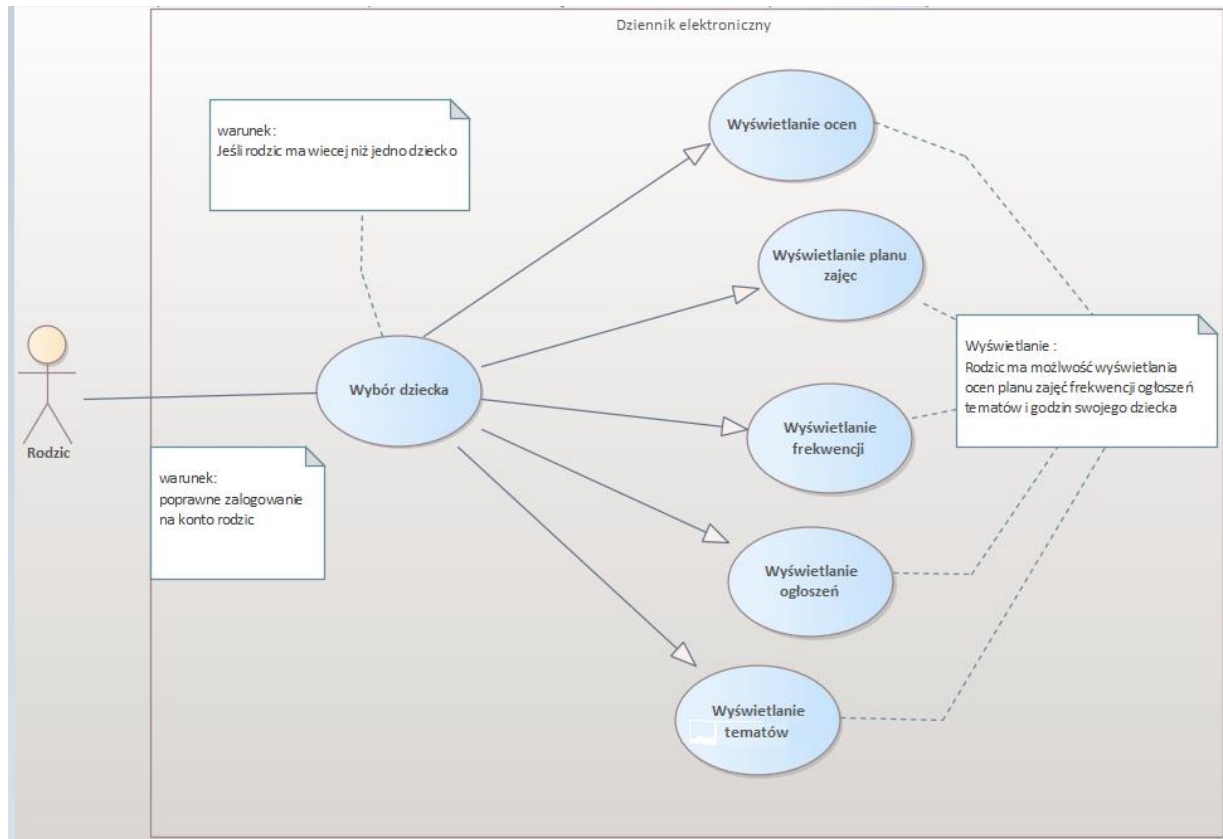


## Rodzic

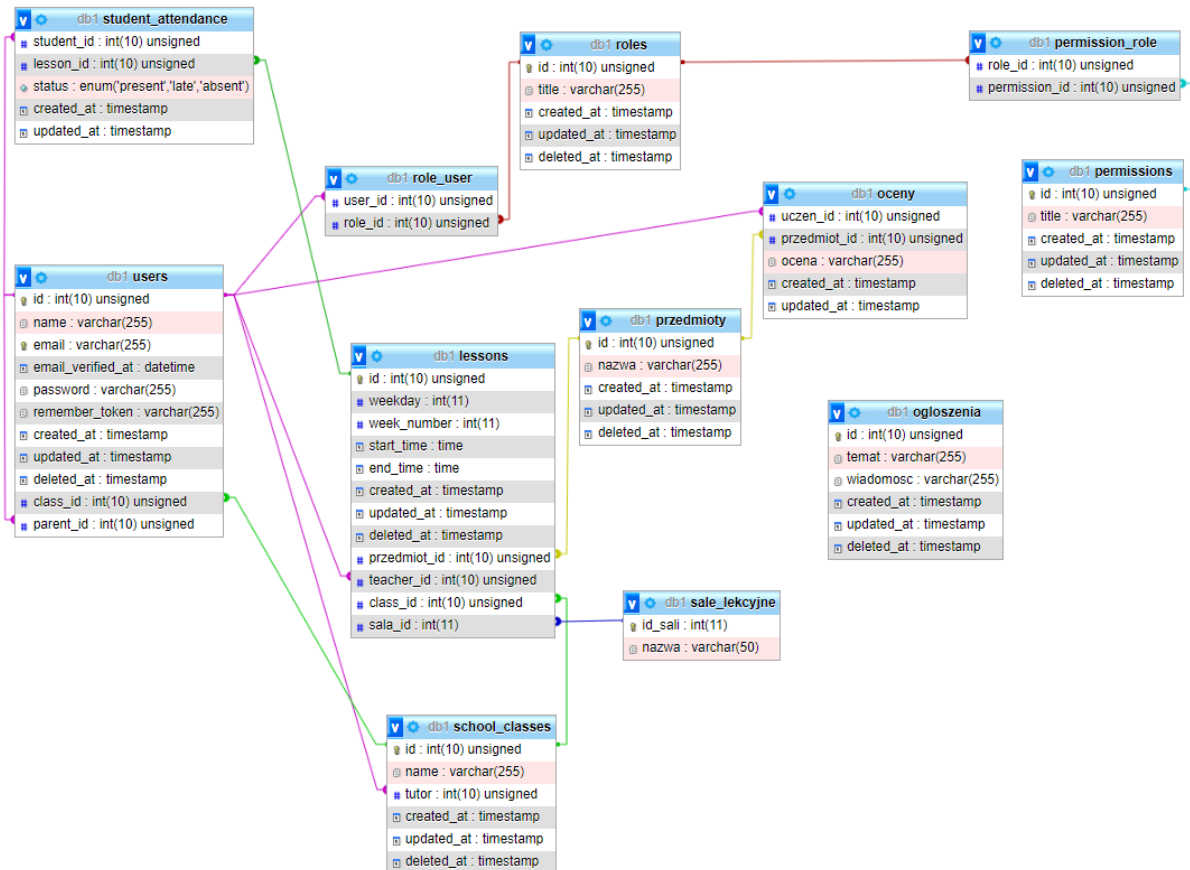




## Uczeń



## Diagram ERD



## Scenariusze przypadków użycia

### Scenariusz nr: 1

**Temat :** Logowanie użytkownika

**Aktorzy :** Administrator, Nauczyciel, Rodzic, Uczeń

**Przebieg :**

1. Użytkownik przystępuje do logowania się w systemie
2. Użytkownik podaje wymagane dane: login i hasło
3. System weryfikuje poprawność wprowadzonych danych

**Zakończenie :**

Po pomyślnej weryfikacji danych przez system użytkownik z każdej grupy ma dostęp do swoich funkcji według diagramu przypadków użycia

**Zakończenie alternatywne :**

System po nie udanej weryfikacji danych wyświetla błąd logowania wraz z komunikatem „hasło i login nie pasują do siebie ” oraz umożliwia użytkownikowi wprowadzenie ponownie hasła i loginu w celu dalszej próby zalogowania się do systemu

### Scenariusz nr: 2

**Temat :** Opcje administratora po poprawnym zalogowaniu

**Aktorzy :** Administrator

**Przebieg :**

1. Po poprawnym zalogowaniu administrator ma dostęp do panelu administracyjnego
2. Administrator ma możliwość wybrania opcji którą chce wykonać z menu :
  - Dodawanie ucznia, nauczyciela, rodzica
  - Usunięcie ucznia, nauczyciela, rodzica
  - Dodawanie przedmiotu
  - Usunięcie przedmiotu
  - Modyfikacja danych
  - Dodanie nauczyciela do przedmiotu
  - Zmiana nauczyciela przedmiotu
  - Dodanie planu zajęć
3. Wybór opcji przez administratora
4. Wprowadzenie danych do wybranej opcji przez administratora
5. Zatwierdzenie danych

**Zakończenie :**

Po poprawnym zalogowaniu administrator ma dostęp do panelu administracyjnego z opcjami widocznymi na diagramie przypadków użycia może kolejno wybierać czynność którą chce podjąć po wybraniu czynności i wprowadzeniu danych może je zapisać

**Zakończenia alternatywne :**

System wyświetla błąd w związku z wyborem opcji

System wyświetla błąd w związku z problemami z zapisem wprowadzonych danych

System wyświetla błąd w związku z problemami wyświetlania danych

**Scenariusz nr: 3**

**Temat :** Opcje nauczyciela po poprawnym zalogowaniu

**Aktorzy :** Nauczyciel

**Przebieg :**

1. Nauczyciel po poprawnym zalogowaniu ma dostęp do panelu administracyjnego dla nauczyciela
2. Nauczyciel ma dostęp do menu z opcjami zgodnymi z diagramem użycia :
  - Wybór zajęć
  - Sprawdzanie obecności
  - Wyświetlanie planu zajęć
  - Wpisywanie ocen
  - Wpisywanie ogłoszeń
  - Wpisywanie tematu
3. Nauczyciel dokonuje wyboru opcji
4. Nauczyciel wprowadza dane do wybranej opcji
5. Zatwierdzenie danych

**Zakończenie :**

Po poprawnym zalogowaniu nauczyciel ma dostęp do panelu nauczyciela z opcjami widocznymi na diagramie przypadków użycia może kolejno wybierać czynność którą chce podjąć po wybraniu czynności i wprowadzeniu danych może je zapisać

**Zakończenia alternatywne :**

System wyświetla błąd w związku z wyborem opcji

System wyświetla błąd w związku z problemami z zapisem wprowadzonych danych

System wyświetla błąd w związku z problemami wyświetlania danych

**Scenariusz nr: 4**

**Temat :** Opcje rodzica po poprawnym zalogowaniu

**Aktorzy :** Rodzic

**Przebieg :**

1. Rodzic po poprawnym zalogowaniu ma dostęp do panelu rodzica
2. Wybór dziecka (jeśli posiada więcej niż jedno)
3. Nauczyciel ma dostęp do menu z opcjami zgodnymi z diagramem użycia :

- Wyświetlanie planu zajęć dziecka
  - Wyświetlanie obecności dziecka
  - Wyświetlanie ocen dziecka
  - Wyświetlanie ogłoszeń dziecka
  - Wyświetlanie tematów
4. Rodzic dokonuje wyboru opcji
  5. Rodzic sprawdza wyświetlone informacje

**Zakończenie :**

Po poprawnym zalogowaniu rodzic ma dostęp do panelu rodzica z opcjami widocznymi na diagramie przypadków użycia może kolejno wybierać czynność którą chce podjąć po wybraniu czynności ma dostęp do wybranych danych

**Zakończenia alternatywne :**

System wyświetla błąd w związku z wyborem opcji

System wyświetla błąd w związku z problemami z wyborem dziecka

System wyświetla błąd w związku z problemami wyświetlania danych

**Scenariusz nr: 5**

**Temat :** Opcje ucznia po poprawnym zalogowaniu

**Aktorzy :** Uczeń

**Przebieg :**

1. Uczeń po poprawnym zalogowaniu ma dostęp do panelu rodzica
2. Uczeń ma dostęp do menu z opcjami zgodnymi z diagramem użycia :
  - Wyświetlanie planu zajęć
  - Wyświetlanie obecności
  - Wyświetlanie ocen
  - Wyświetlanie ogłoszeń
  - Wyświetlanie tematów
3. Uczeń dokonuje wyboru opcji
4. Uczeń sprawdza wyświetlone informacje

**Zakończenie :**

Po poprawnym zalogowaniu uczeń ma dostęp do panelu ucznia z opcjami widocznymi na diagramie przypadków użycia może kolejno wybierać czynność którą chce podjąć po wybraniu czynności ma dostęp do wybranych danych

**Zakończenia alternatywne :**

System wyświetla błąd w związku z wyborem opcji

System wyświetla błąd w związku z problemami wyświetlania danych

### Scenariusz nr: 6

**Temat :** Wylogowanie użytkownika

**Aktorzy :** Administrator, Nauczyciel, Rodzic, Uczeń

**Przebieg :**

1. Użytkownik przystępuje do wylogowania się w systemie
2. Wybiera opcje wyloguj
3. System wylogowywane się z konta użytkownika

**Zakończenie :**

Po pomyślnym wylogowaniu użytkownik ma możliwość zalogowania się ponownie lub na inne konto

**Zakończenie alternatywne :**

System wyświetla błąd o nie powodzeniu wylogowania

### Scenariusz nr: 7

**Temat :** Dodawanie nauczyciela ucznia i rodzica

**Aktorzy :** Administrator

**Przebieg :**

1. Administrator wybiera opcje związaną z dodawaniem użytkowników
2. Administrator wpisuje dane dotyczące osobowe użytkownika oraz wybiera rodzaj użytkownika czy jest to uczeń rodzic czy nauczyciel
3. Administrator dokonuje zapisu wprowadzonych danych
4. Admirator może powrócić do wyboru innych zajęć lub kontynuować opcje dodawania użytkowników

**Zakończenie :**

Po pomyślnym wprowadzeniu danych dane zostają zapisane w bazie danych

**Zakończenie alternatywne :**

System wyświetla błąd o nie powodzeniu zasiania wprowadzonych danych

**Warunki wejścia :**

Aby administrator mógł dodać nowego użytkownika musi mieć wymagane dane osobowe danego użytkownika

## Scenariusz nr: 8

**Temat :** Usuwanie użytkownika

**Aktorzy :** Administrator

**Przebieg :**

1. Administrator wybiera opcje związaną z usuwaniem użytkownika
2. Admirator wybiera grupę użytkowników np. nauczyciele
3. Po odnalezieniu wybranego użytkownika admirator wybiera opcje usuń
4. Użytkownik zostaje usunięty
5. Admirator może powrócić do wyboru innych zajęć lub kontynuować opcje usuwania użytkowników

**Zakończenie :**

Użytkownik zostaje usunięty z danej grupy

**Zakończenie alternatywne :**

System wyświetla błąd związany z problemem usunięcia użytkownika

## Scenariusz nr: 9

**Temat :** Modyfikacja danych

**Aktorzy :** Administrator

**Przebieg :**

1. Administrator wybiera opcje związaną z modyfikacją danych
2. Administrator wybiera dane które chce zmodyfikować np. obecność czy frekwencje
3. Administrator wpisuje dane do zamiany
4. Administrator dokonuje zapisu wprowadzonych danych
5. Admirator może powrócić do wyboru innych zajęć lub kontynuować opcje modyfikowania danych

**Zakończenie :**

Po pomyślnym wprowadzeniu danych dane zostają zapisane w bazie danych

**Zakończenie alternatywne :**

System wyświetla błąd o nie powodzeniu zapisania wprowadzonych danych

**Warunki wejścia :**

Aby administrator mógł modyfikować dane muszą zostać wprowadzone wcześniej jakieś dane , wtedy na wcześniej wprowadzonych danych administrator może dokonać zmian

### Scenariusz nr: 10

**Temat :** Dodanie nauczyciela do przedmiotu

**Aktorzy :** Administrator

**Przebieg :**

1. Administrator wybiera opcje związaną z przypisywaniem nauczyciela do przedmiotu
2. Administrator wybiera nauczyciela i przypisuje go do przedmiotu
3. Administrator zatwierdza zmianę i zapisuje stan
4. Administrator może powrócić do wyboru innych zajęć lub kontynuować opcje przypisywania nauczyciela do przedmiotu

**Zakończenie :**

Po pomyślnym przypisaniu nauczyciela do przedmiotu nauczyciel jest przypisany do przedmiotu

**Zakończenie alternatywne :**

System wyświetla błąd o nie powodzeniu zapisania wprowadzonych zmian

**Warunki wejścia :**

Aby można było dodać nauczyciela do przedmiotu najpierw nauczyciel musi istnieć w bazie jeden nauczyciel może uczyć wiele przedmiotów

### Scenariusz nr: 11

**Temat :** Usuwanie przedmiotu

**Aktorzy :** Administrator

**Przebieg :**

1. Administrator wybiera opcje związaną z usuwaniem przedmiotu
2. Administrator wybiera przedmiot który chce usunąć
3. Administrator zatwierdza zmianę i zapisuje stan , przedmiot zostaje usunięty
4. Administrator może powrócić do wyboru innych zajęć lub kontynuować opcje usuwania przedmiotu

**Zakończenie :**

Przedmiot zostaje usunięty

**Zakończenie alternatywne :**

System wyświetla błąd o nie powodzeniu usunięcia przedmiotu

**Warunki wejścia :**

Aby usunąć przedmiot najpierw przedmiot musi istnieć w bazie danych

### Scenariusz nr: 12

**Temat :** Dodanie planu zajęć

**Aktorzy :** Administrator

**Przebieg :**

1. Administrator wybiera opcje związaną z dodawaniem planu zajęć
2. Administrator wybiera plan zajęć i dodaje przedmioty i godziny oraz klasę
3. Administrator zatwierdza zmianę i zapisuje stan
4. Administrator może powrócić do wyboru innych zajęć lub kontynuować opcje dodawania planu zajęć

**Zakończenie :**

Po pomyślnym dodaniu plan zostaje zapisany do bazy danych

**Zakończenie alternatywne :**

System wyświetla błąd o nie poprawnym zapisaniu danych

### Scenariusz nr: 13

**Temat :** Sprawdzanie obecności

**Aktorzy :** Nauczyciel

**Przebieg :**

1. Nauczyciel wybiera opcje sprawdzania obecności
2. Nauczyciel wybiera klasę do sprawdzenia obecności i wprowadza status ucznia czy jest obecny czy też nie
3. Nauczyciel zatwierdza zmianę i zapisuje dane
4. Nauczyciel może powrócić do wyboru innych zajęć

**Zakończenie :**

Po pomyślnym zapisaniu frekwencja trafia do bazy danych

**Zakończenie alternatywne :**

System wyświetla błąd o nie poprawnym zapisaniu danych

### Scenariusz nr: 14

**Temat :** Wyświetlanie planu zajęć

**Aktorzy :** Nauczyciel

**Przebieg :**

1. Nauczyciel wybiera opcje wyświetlanie planu zajęć
2. Nauczyciel sprawdza swój plan zajęć
3. Nauczyciel może powrócić do wyboru innych zajęć



**Zakończenie :**

Wyświetla się plan zajęć nauczyciela

**Zakończenie alternatywne :**

System wyświetla błąd

**Scenariusz nr: 15**

**Temat :** Wpisywanie ocen

**Aktorzy :** Nauczyciel

**Przebieg :**

1. Nauczyciel wybiera opcje wpisywania ocen
2. Nauczyciel wybiera klasę do wpisywania ocen i wprowadza ocenę dla ucznia z danego przedmiotu którego uczy
3. Nauczyciel zapisuje wprowadzoną ocenę
4. Nauczyciel może powrócić do wyboru innych zajęć

**Zakończenie :**

Po pomyślnym zapisaniu ocena trafia do bazy danych

**Zakończenie alternatywne :**

System wyświetla błąd o nie poprawnym zapisaniu danych

**Scenariusz nr: 16**

**Temat :** Wpisywanie ogłoszeń

**Aktorzy :** Nauczyciel

**Przebieg :**

1. Nauczyciel wybiera opcje wpisywania ogłoszeń
2. Nauczyciel zapisuje wprowadzone ogłoszenie
3. Nauczyciel może powrócić do wyboru innych zajęć

**Zakończenie :**

Po pomyślnym zapisaniu ogłoszenie trafia do bazy danych

**Zakończenie alternatywne :**

System wyświetla błąd o nie poprawnym zapisaniu danych

### Scenariusz nr: 17

**Temat :** Wpisywanie tematów

**Aktorzy :** Nauczyciel

**Przebieg :**

1. Nauczyciel wybiera opcje wpisywania tematów
2. Nauczyciel wybiera klasę do wpisywania tematów i wprowadza temat
3. Nauczyciel zapisuje wprowadzony temat
4. Nauczyciel może powrócić do wyboru innych zajęć

**Zakończenie :**

Po pomyślnym zapisaniu temat trafia do bazy danych

**Zakończenie alternatywne :**

System wyświetla błąd o nie poprawnym zapisaniu danych

### Scenariusz nr: 18

**Temat :** Wyświetlanie ocen

**Aktorzy :** Rodzic

**Przebieg :**

1. Rodzic przechodzi do sprawdzenia ocen dziecka do którego jest przypisany (jeśli ma więcej niż jedno dziecko to po zalogowaniu dokonuje wyboru dziecka którego dane będzie chciał sprawdzić )
2. Rodzic ma możliwość sprawdzenia ocen swojego dziecka
3. Rodzic może powrócić do wyboru innych zajęć

**Zakończenie :**

Wyświetlenie ocen swojego dziecka

**Zakończenie alternatywne :**

W przypadku braku ocen system nie wyświetli ocen

System wyświetla błąd związany z wyświetleniem ocen dziecka

### Scenariusz nr: 19

**Temat :** Wyświetlanie planu zajęć

**Aktorzy :** Rodzic

**Przebieg :**

1. Rodzic przechodzi do sprawdzenia planu zajęć dziecka do którego jest przypisany (jeśli ma więcej niż jedno dziecko to po zalogowaniu dokonuje wyboru dziecka którego dane będzie chciał sprawdzić )
2. Rodzic ma możliwość sprawdzenia planu zajęć swojego dziecka
3. Rodzic może powrócić do wyboru innych zajęć

**Zakończenie :**

Wyświetlenie planu zajęć swojego dziecka

**Zakończenie alternatywne :**

System wyświetla błąd związany z wyświetleniem planu zajęć dziecka

### Scenariusz nr: 20

**Temat :** Wyświetlanie frekwencji

**Aktorzy :** Rodzic

**Przebieg :**

1. Rodzic przechodzi do sprawdzenia frekwencji dziecka do którego jest przypisany (jeśli ma więcej niż jedno dziecko to po zalogowaniu dokonuje wyboru dziecka którego dane będzie chciał sprawdzić )
2. Rodzic ma możliwość sprawdzenia frekwencji swojego dziecka
3. Rodzic może powrócić do wyboru innych zajęć

**Zakończenie :**

Wyświetlenie frekwencji swojego dziecka

**Zakończenie alternatywne :**

System wyświetla błąd związany z wyświetleniem frekwencji dziecka

## Scenariusz nr: 21

**Temat :** Wyświetlanie ogłoszeń

**Aktorzy :** Rodzic

**Przebieg :**

1. Rodzic przechodzi do sprawdzenia ogłoszenia dla dziecka do którego jest przypisany (jeśli ma więcej niż jedno dziecko to po zalogowaniu dokonuje wyboru dziecka którego dane będzie chciał sprawdzić )
2. Rodzic ma możliwość sprawdzenia ogłoszeń swojego dziecka
3. Rodzic może powrócić do wyboru innych zajęć

**Zakończenie :**

Wyświetlenie ogłoszeń swojego dziecka

**Zakończenie alternatywne :**

W przypadku braku ogłoszeń system nie wyświetli ogłoszeń

System wyświetla błąd związany z wyświetleniem ogłoszeń dziecka

## Scenariusz nr: 22

**Temat :** Wyświetlanie tematów

**Aktorzy :** Rodzic

**Przebieg :**

1. Rodzic przechodzi do sprawdzenia tematów dziecka do którego jest przypisany (jeśli ma więcej niż jedno dziecko to po zalogowaniu dokonuje wyboru dziecka którego dane będzie chciał sprawdzić )
2. Rodzic ma możliwość sprawdzenia tematów swojego dziecka
3. Rodzic może powrócić do wyboru innych zajęć

**Zakończenie :**

Wyświetlenie tematów swojego dziecka

**Zakończenie alternatywne :**

System wyświetla błąd związany z wyświetleniem tematów dziecka

### Scenariusz nr: 23

**Temat :** Wyświetlanie ocen

**Aktorzy :** Uczeń

**Przebieg :**

1. Uczeń przechodzi do sprawdzenia ocen
2. Uczeń ma możliwość sprawdzenia swoich ocen
3. Uczeń może powrócić do wyboru innych zajęć

**Zakończenie :**

Wyświetlenie ocen ucznia

**Zakończenie alternatywne :**

W przypadku braku ocen system nie wyświetli ocen

System wyświetla błąd związany z wyświetleniem ocen ucznia

### Scenariusz nr: 24

**Temat :** Wyświetlanie planu zajęć

**Aktorzy :** Uczeń

**Przebieg :**

1. Uczeń przechodzi do sprawdzenia planu zajęć
2. Uczeń ma możliwość sprawdzenia swojego planu zajęć
3. Uczeń może powrócić do wyboru innych zajęć

**Zakończenie :**

Wyświetlenie planu zajęć ucznia

**Zakończenie alternatywne :**

System wyświetla błąd związany z wyświetleniem planu zajęć ucznia

### Scenariusz nr: 25

**Temat :** Wyświetlanie frekwencji

**Aktorzy :** Uczeń

**Przebieg :**

1. Uczeń przechodzi do sprawdzenia frekwencji
2. Uczeń ma możliwość sprawdzenia swojej frekwencji zajęć
3. Uczeń może powrócić do wyboru innych zajęć

**Zakończenie :**

Wyświetlenie frekwencji ucznia

**Zakończenie alternatywne :**

System wyświetla błąd związany z wyświetleniem frekwencji ucznia

**Scenariusz nr: 26**

**Temat :** Wyświetlanie ogłoszeń

**Aktorzy :** Uczeń

**Przebieg :**

1. Uczeń przechodzi do sprawdzenia ogłoszeń
2. Uczeń ma możliwość sprawdzenia swoich ogłoszeń
3. Uczeń może powrócić do wyboru innych zajęć

**Zakończenie :**

Wyświetlenie ogłoszeń ucznia

**Zakończenie alternatywne :**

W przypadku braku ocen system nie wyświetli ogłoszeń

System wyświetla błąd związany z wyświetleniem ogłoszeń ucznia

**Scenariusz nr: 27**

**Temat :** Wyświetlanie tematów

**Aktorzy :** Uczeń

**Przebieg :**

1. Uczeń przechodzi do sprawdzenia tematów
2. Uczeń ma możliwość sprawdzenia swoich tematów
3. Uczeń może powrócić do wyboru innych zajęć

**Zakończenie :**

Wyświetlenie tematów ucznia

**Zakończenie alternatywne :**

System wyświetla błąd związany z wyświetleniem tematów ucznia

## Opis technologii

**JavaScript** - jest to skryptowy język, który umożliwia obsługę dynamicznego tworzenia treści na stronie internetowej, kontrolowanie multimediów, animację obrazów i prawie wszystko inne .

**MySQL** – ogólnodostępny system zarządzania relacyjnymi bazami danych. Był pisany raczej z myślą o szybkości niż kompatybilności ze standardem SQL – przez dłuższy czas MySQL nie obsługiwał nawet transakcji. Wprowadza również swoje rozszerzenia i nowe elementy języka. Serwer MySQL dostępny jest dla wszystkich popularnych platform systemowych i różnorodnych architektur procesorów.

**HTML** – pozwala opisać strukturę informacji zawartych wewnątrz strony internetowej, nadając odpowiednie znaczenie semantyczne poszczególnym fragmentom tekstu – formując hiperłącza, akapity, nagłówki, listy – oraz osadza w tekście dokumentu obiekty plikowe np. multimedia bądź elementy baz danych np. interaktywne formularze danych.

**CSS (Kaskadowe arkusze stylów)** – język służący do opisu formy prezentacji (wyświetlania) stron WWW, został stworzony w celu odseparowania struktury dokumentu od formy jego prezentacji.

**Domena internetowa** – ciąg identyfikacyjny systemu Domain Name System (DNS), który określa zakres autonomii administracyjnej, uprawnień lub kontroli w Internecie. Nazwa domeny składa się z co najmniej jednej części, które (nazywane technicznie etykietami) są umieszczone w pewnym poddrzewie struktury DNS. Etykiety te są łączone i rozdzielane kropkami.

**GitHub** – hostingowy serwis internetowy przeznaczony dla projektów programistycznych wykorzystujących system kontroli wersji Git. Stworzony został przy wykorzystaniu frameworka Ruby on Rails i języka Erlang. Serwis działa od kwietnia 2008 roku. GitHub udostępnia darmowy hosting programów open source i prywatnych repozytoriów.

**phpMyAdmin** – narzędzie służące do łatwego zarządzania bazą danych MySQL, napisane w języku PHP. Oprogramowanie wydawane jest na licencji GNU General Public License i umożliwia między innymi tworzenie/usuwanie baz danych, dodawanie/kasowanie relacji oraz edycję ich struktury i zawartości.

**jQuery** – lekka biblioteka programistyczna przeznaczona dla języka JavaScript. jQuery jest najpopularniejszą z obecnie używanych bibliotek JavaScript'owych. Biblioteka ta jest darmowa i oparta na licencji zgodnej z definicją otwartego oprogramowania (open-source).

**Framework** – szkielet do budowy aplikacji. Definiuje on strukturę aplikacji oraz ogólny mechanizm jej działania, a także dostarcza zestaw komponentów i bibliotek ogólnego przeznaczenia do wykonywania określonych zadań.

**Laravel** – framework do aplikacji internetowych napisany w języku PHP bazujący na wzorcu architektonicznym Model-View-Controller.

**Composer** – system zarządzania pakietami dla języka PHP, dostępny jako aplikacja wiersza poleceń, która dostarcza i standaryzuje format zarządzania zależnościami skryptami i bibliotekami.

**Bootstrap** - biblioteka CSS, rozwijana przez programistów Twittera, wydawany na licencji MIT. Zawiera zestaw przydatnych narzędzi ułatwiających tworzenie interfejsu graficznego stron oraz aplikacji internetowych.

**PHP** – interpretowany, skryptowy język programowania zaprojektowany do generowania stron internetowych i budowania aplikacji webowych w czasie rzeczywistym.



## Implementacja bazy danych – migracje

### Tabela users

Poniższy wycinek kodu przedstawia migrację tworzącą w bazie danych tabelę users. Id users będzie wykorzystywane jako klucz obcy w innych tabelach (np przypisanie roli użytkownikowi w tabeli role\_user).

```
class CreateUsersTable extends Migration
{
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('email')->unique();
            $table->datetime('email_verified_at')->nullable();
            $table->string('password');
            $table->string('remember_token')->nullable();
            $table->timestamps();
            $table->softDeletes();
        });
    }
}
```

### Tabela roles

```
class CreateRolesTable extends Migration
{
    public function up()
    {
        Schema::create('roles', function (Blueprint $table) {
            $table->increments('id');
            $table->string('title')->nullable();
            $table->timestamps();
            $table->softDeletes();
        });
    }
}
```

## Tabela role\_user

Poniższy kod stanowi implementację tabeli role\_user. Jest to tabela przyporządkowująca danemu użytkownikowi rolę, którą będzie pełnił po jego utworzeniu. Kolumna 'user\_id' to klucz obcy z tabeli users(id), natomiast kolumna 'role\_id' to klucz obcy odnoszący się do tabeli role(id).

```
class CreateRoleUserPivotTable extends Migration
{
    public function up()
    {
        Schema::create('role_user', function (Blueprint $table) {
            $table->unsignedInteger('user_id');
            $table->foreign('user_id', 'user_id_fk_1001484')->
>references('id')
            ->on('users')->onDelete('cascade');
            $table->unsignedInteger('role_id');
            $table->foreign('role_id', 'role_id_fk_1001484')->
>references('id')
            ->on('roles')->onDelete('cascade');
        });
    }
}
```

## Tabela lesson

Tworzenie tabeli lesson, zawierającej id lekcji, numer dnia, numer tygodnia nauki, czas rozpoczęcia i zakończenia lekcji oraz id przedmiotu. Warto dodać, że plan lekcji w głównej mierze jest oparty na danych z tabeli Lessons, więc struktura musiała być dobrze przemyślana, ze względu na skomplikowane działanie wyświetlania planu lekcji.

```
class CreateLessonsTable extends Migration
{
    public function up()
    {
        Schema::create('lessons', function (Blueprint $table) {
            $table->increments('id');
            $table->integer('weekday');
            $table->integer('week_number')->nullable();
            $table->time('start_time');
            $table->time('end_time');
            $table->timestamps();
            $table->softDeletes();
            $table->unsignedInteger('przedmiot_id')->nullable();
        });
    }
}
```

## Tabela school\_classes

Tabela school\_classes to tabela odpowiedzialna za budowę struktury poszczególnych klas. Każda klasa ma swoje id, nazwę oraz wychowawcę. Wychowawca to klucz obcy z tabeli users.

```
class CreateSchoolClassesTable extends Migration
{
    public function up()
    {
        Schema::create('school_classes', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->unsignedInteger('tutor')->nullable();
            $table->timestamps();
            $table->softDeletes();
        });

        Schema::table('school_classes', function (Blueprint $table){
            $table->foreign('tutor')->references('id')->on('users');
        });
    }
}
```

Utworzenie relacji pomiędzy tabelami lessons-users a także lessons-schoolclasses.

```
class AddRelationshipFieldsToLessonsTable extends Migration
{
    public function up()
    {
        Schema::table('lessons', function (Blueprint $table) {
            $table->unsignedInteger('teacher_id');
            $table->foreign('teacher_id', 'teacher_fk_1001496')
                ->references('id')->on('users');
            $table->unsignedInteger('class_id');
            $table->foreign('class_id', 'class_fk_1001508')-
>references('id')
                ->on('school_classes');
        });
    }
}
```

Utworzenie relacji pomiędzy tabelami `users` oraz `school_classes`, a także dodanie kolumny `parent_id` przypisującą uczniowi rodzica. Jeśli rola to nauczyciel, rodzic lub administrator, to pole zostaje puste.

```
class AddRelationshipFieldsToUsersTable extends Migration
{
    public function up()
    {
        Schema::table('users', function (Blueprint $table) {
            $table->unsignedInteger('class_id')->nullable();
            $table->foreign('class_id', 'class_fk_1001550')->
>references('id')->on('school_classes');
            $table->unsignedInteger('parent_id')->nullable();
            $table->foreign('parent_id')->references('id')->on('users');
        });
    }
}
```

#### Tabela `student_attendance` (frekwencja)

Tworzenie tabeli `student_attendance` (obecność). Składają się na nią kolumny:

- `student_id` – klucz obcy z tabeli `users(id)`
- `lesson_id` – klucz obcy z tabeli `lessons(id)`
- `status` – typu ENUM (1 – obecny, 2 – spóźniony, 3 – nieobecny)

```
class CreateStudentAttendanceTable extends Migration
{
    public function up()
    {
        Schema::create('student_attendance', function (Blueprint $table) {
            $table->unsignedInteger('student_id');
            $table->foreign('student_id')->references('id')->on('users');
            $table->unsignedInteger('lesson_id');
            $table->foreign('lesson_id')->references('id')->on('lessons');
            $table->enum('status', ['present', 'late', 'absent']);
            $table->timestamps();
        });
    }
}
```

## Tabela przedmioty

Tworzenie tabeli przedmioty składającej się z id oraz nazwy, oraz stworzenie relacji pomiędzy tabelą z wykorzystaniem lessons.przedmiot\_id oraz przedmioty.id.

```
class CreatePrzedmiotyTable extends Migration
{
    public function up()
    {
        Schema::create('przedmioty', function (Blueprint $table) {
            $table->increments('id');
            $table->string('nazwa');
            $table->timestamps();
            $table->softDeletes();
        });

        Schema::table('lessons', function (Blueprint $table) {
            $table->foreign('przedmiot_id')->references('id')
                ->on('przedmioty');
        });
    }
}
```

## Tabela ogłoszenia

Tworzenie tabeli ogłoszenia zawierającą id, temat oraz wiadomość. Ogłoszenia mogą tworzyć jedynie role mające uprawnienie 'announcement\_create'

```
class CreateOgloszeniaTable extends Migration
{
    public function up()
    {
        Schema::create('ogloszenia', function (Blueprint $table) {
            $table->increments('id');
            $table->string('temat');
            $table->string('wiadomosc');
            $table->timestamps();
            $table->softDeletes();
        });
    }
}
```

## Tabela oceny

Tabela oceny składa się z uczen\_id (klucz obcy z users(id)), przedmiot\_id (klucz obcy z przedmioty(id)), ocena. W związku z tym, że funkcjonalność wpisywania ocen nie została osiągnięta w naszej witrynie, struktura tabeli została utworzona zgodnie z projektem bazy danych, utworzonym przed rozpoczęciem implementacji.

```
class CreateOcenyTable extends Migration
{
    public function up()
    {
        Schema::create('oceny', function (Blueprint $table) {
            $table->unsignedInteger('uczen_id');
            $table->unsignedInteger('przedmiot_id');
            $table->string('ocena')->nullable();
            $table->foreign('uczen_id')->references('id')->on('users');
            $table->foreign('przedmiot_id')->references('id')
                ->on('przedmioty');
            $table->timestamps();
        });
    }
}
```

## Komunikacja z bazą danych mysql

Podczas tworzenia witryny, do komunikacji z bazą danych wykorzystaliśmy wbudowane narzędzie Eloquent ORM pozwalające na intuicyjny i szybki dostęp do danych w bazie mysql, oraz zapytania typu RAW, będące bezpośrednimi zapytaniami do bazy danych z poziomu języka PHP. Bardzo ważnym elementem tworzenia zapytań do bazy danych dla ORM jest utworzenie Modelu, który jest używany do interakcji z daną tabelą w bazie.

Przykładem wykorzystania ORM jest tworzenie ogłoszenia z poziomu formularza w panelu administracyjnym zawartym na stronie.

```
public function store(StoreOgloszeniaRequest $request)
{
    $ogloszenia = Ogloszenia::create($request->all());

    return redirect()->route('admin.ogloszenia.index');
}
```

Powyższy kod wprowadza dane do tabeli ogłoszenia w oparciu o model Ogłoszenia, który wygląda następująco:

```
<?php
namespace App;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;

class Ogloszenia extends Model
{
    use SoftDeletes;
    public $table = 'ogloszenia';
    protected $dates = [
        'created_at',
        'updated_at',
        'deleted_at',
    ];
    protected $fillable = [
        'temat',
        'wiadomosc',
        'updated_at',
        'deleted_at',
    ];
}
```

Jak wiemy klasa laravela `Illuminate\Http\Request` zapewnia obiektowy sposób interakcji z bieżącym żądaniem HTTP obsługiwany przez aplikację, a także pobieranie danych wejściowych (w tym ciasteczek i plików) które zostały przesłane wraz z żądaniem.

Dla powyższego przykładu w katalogu `App\Http\Request` została utworzona klasa `StoreOgloszeniaRequest.php`

```
<?php

namespace App\Http\Requests;
use App\Ogloszenia;
use Gate;
use Illuminate\Foundation\Http\FormRequest;
use Symfony\Component\HttpFoundation\Response;

class StoreOgloszeniaRequest extends FormRequest
{
    public function authorize()
    {
        return true;
    }
    public function rules()
    {
        return [
            'temat' => [
                'required'],
            'wiadomosc' => [
                'required'],
        ];
    }
}
```

Ostatecznie dane przekazane do funkcji z poziomu formularza zostają przekazane przez funkcję `store` oraz obiektu `$request` do tabeli ogłoszenia zawartej w bazie danych.

Innym typem komunikacji z bazą danych jest wykorzystanie struktury RAW, która ułatwiła nam usuwanie rekordów z bazy.

Przykład wykorzystania znajduje się w kontrolerach praktycznie każdej funkcjonalności.



W poniższym przykładzie funkcja usuwająca rekord z bazy (usuwanie użytkownika).

```
public function destroy(User $user)
{
    abort_if(Gate::denies('user_delete'), Response::HTTP_FORBIDDEN, '403 Forbidden');

    DB::connection('mysql')->delete(DB::raw("UPDATE users SET deleted_at=NOW() WHERE id= " .
    $user->id . " ; "));

    return back();
}
```

Argumentem dla funkcji jest id użytkownika przekazane po wciśnięciu przycisku usuń w widoku. Funkcja dodatkowo sprawdza, czy dany użytkownik posiada uprawnienia zezwalające mu na usuwanie użytkowników. Korzystając z id zostaje również wskazany konkretny użytkownik w bazie danych. Dzięki możliwości użycia zmiennej pochodzącej z języka php w zapytaniu SQL, po klauzuli WHERE, id porównane z wartością zmiennej powoduje wybranie konkretnego rekordu z bazy danych oraz zaktualizowanie jego rekordu w kolumnie deleted\_at.

## Harmonogram Pracy

Paweł i Piotr

- Zapoznanie się z środowiskiem pracy w Frameworku Laravel → 10h

Paweł

- Rozpoczęcie tworzenia bazy danych i połączenia ze stroną → 15h  
Utworzenie bazy oraz sprawdzenie jej połączenia ze stroną
- Tworzenie poszczególnych tabel i relacji → 15h  
Utworzenie na podstawie diagramu ERD potrzebnych tabel i relacji pomiędzy nimi
- Wprowadzenie danych do tabel → 8h  
Wprowadzenie przykładowych danych do tabel użytkowników przedmiotów klas
- Stworzenie zapytań odpowiedzialnych za logowanie na konta użytkowników → 15h

Piotr

- Tworzenie panelu logowania → 12h
- Tworzenie kolejnych widoków dla konta nauczyciel → 14h  
Utworzenie opcji dla konta nauczyciela
- Tworzenie kolejnych widoków dla konta rodzic → 14h  
Utworzenie opcji dla konta rodzica
- Tworzenie kolejnych widoków dla konta uczeń → 14h  
Utworzenie opcji dla konta ucznia
- Tworzenie kolejnych widoków dla konta administrator → 14h  
Utworzenie opcji dla konta administratora

Paweł i Piotr

- Poprawki związane z bazą i aplikacją strony → 12h  
Szukanie potencjalnych błędów w działaniu strony web
- Testy działania → 8h

**Przewidywany czas pracy: 190 h**

## Lista zadań oraz estymacja czasowa

### Tydzień 04.05.2022

Piotr Bożek: 8h

-Wybór i instalacja środowiska (Laravel Composer phpmyadmin ) oraz założenie projektu na GitHubie i dodanie członków

Paweł Cetnarowski: 7h

-Wybór i instalacja środowiska (Laravel Composer phpmyadmin )

### Tydzień 11.05.2022

Piotr Bożek: 22h

-utworzenie widoku administratora z możliwością dodawania użytkowników

-model lekcji + kontroler + migracje + przykładowe dane + dodanie danych do seederów

Paweł Cetnarowski: 22h

-utworzenie podziału ról dla administratora (jaką rolę pełni użytkownicy)

-model klasy + kontroler + migracje + przykładowe dane + dodanie danych do seederów

### Tydzień 18.05.2022

Paweł Cetnarowski: 25h

- Wykorzystanie uprawnień w menu, w celu wyświetlania konkretnych opcji dla użytkowników o danej roli

- Podgląd i edycja klas

- Kreator tworzenia lekcji

-przypisanie rodzica do ucznia

Piotr Bożek: 25h

- Dodawanie/usuwanie użytkownika z bazy

-nadawanie uprawnień

-przypisanie go do klasy (jeśli jest uczniem), nadawanie użytkownikom roli.

-wyświetlanie poszczególnych użytkowników

## **Tydzień 25.05.2022**

Piotr Bożek: 12h

- Dodanie w panelu zarządzania opcji: użytkownicy, nauczyciele, uczniowie, które odpowiadają za dodawanie poszczególnych typów kont oraz wyświetlanie kont i danych użytkowników
- Dodawanie uczniów oraz przypisywanie ról oraz danych dla poszczególnych kont. Jeżeli użytkownik jest uczniem, przypisywanie do klasy
- Przypisanie rodzica do ucznia

Paweł Cetnarowski: 18h

- Wyświetlanie menu według konkretnych uprawnień - odwołanie do uprawnień za pomocą dyrektywy can (docelowo zostanie stworzony widok umożliwiający administratorowi edycję uprawnień dla poszczególnych ról - uprawnienia przypisane są do roli)
- Podgląd i edycja klas - widok uczniów przypisanych do danej klasy, dodawanie nowej klasy
- Tworzenie lekcji: przypisanie klasy
- Przypisywanie nauczyciela przedmiotu

## **Tydzień 01.06.2022**

Paweł Cetnarowski: 26h

- utworzenie wprowadzania ocen i przypisywanie ich dla danego ucznia oraz przedmiotu,
- dodanie wychowawcy klasy
- widok edycji uprawnień dla poszczególnych ról, dodawanie nowych uprawnień
- sprawdzanie obecności

Piotrek Bożek: 28h

- edycja danych istniejących użytkowników
- utworzenie widoku do wyświetlania danych osobowych utworzonych kąt
- utworzenie tabel przedmiotów i tematów - uzupełnienie tabeli
- utworzenie widoku planu zajęć dla poszczególnych klas

## Tydzień 08.06.2022

Paweł Cetnarowski: 20h

-Dodanie widoku ogłoszeń

-Dodanie widoku sprawdzania obecności

Piotrek Bożek: 20h

-Naprawienie błędu wyświetlania planu znając

- Wysiedlanie planu zajęci odpowiednio przypisanym użytkownikom

## Bilans

### W naszej pracy udało się zrealizować :

- Budowa bazy danych
- Utworzenie połączenia między bazą danych a stroną internetową
- Szablon logowania
- Szablon panelu administratora
  - Panel zarządzania
  - Panel tworzenia, usuwania oraz edycji uprawnień
  - Panel tworzenia, usuwania, przypisywania oraz edycji ról
  - Panel tworzenia, usuwania oraz edycji użytkowników
  - Panel dodawania ucznia
  - Panel wyświetlania kont użytkowników na ich rodzaj Nauczyciele, Rodzice, Uczniowie, Administratorzy
  - Panel tworzenia, usuwania oraz edycji klas
  - Panel tworzenia, usuwania oraz modyfikacji lekcji
  - Panel ocen
  - Panel generujący plan zajęć na podstawie utworzonych lekcji dla poszczególnych klas
  - Panel tworzenia ogłoszeń
- Szablon panelu nauczyciela
  - Panel oceny
  - Panel planu zajęć nauczyciela
  - Panel ogłoszeń
- Szablon panelu ucznia
  - Panel oceny
  - Panel planu zajęć nauczyciela
  - Panel ogłoszeń

- Szablon panelu rodzica
  - Panel oceny
  - Panel planu zajęć nauczyciela
  - Panel ogłoszeń

#### Nie wykonaliśmy :

- Wprowadzania ocen do bazy danych , tematów
  - Dla nauczycieli
  - Uczniów
- Wprowadzania obecności do bazy danych
  - Dla nauczycieli
  - Uczniów

## Wnioski

Tworzenie dziennika elektronicznego przebiegło pomyślnie. Udało nam się zrealizować większość zamierzonych celów, niestety części z nich nie udało się wykonać przez co musieliśmy porzucić pewne funkcjonalności na rzecz ważniejszych funkcji. Pracując wspólnie podczas realizacji projektu zapoznaliśmy się z nowym dla nas frameworkiem Laravel - z dnia na dzień poszerzając wiedzę na jego temat. Zdecydowaliśmy się na stworzenie wyglądu strony z pomocą Bootstrapa oraz elementów jquery, javascript. Wierzmy, że po zakończeniu pracy projektowej w dalszym ciągu będziemy poszerzać swoją wiedzę na temat frameworków oraz z nich korzystać. Podczas realizacji projektu nauczyliśmy się również pracy z bazą danych, tworzeniem zapytań SQL typu RAW oraz wykorzystywać wbudowane już funkcje komunikacji z bazą danych ORM.

Projekt dziennika elektronicznego stanowi dla nas nowe doświadczenia, powodem tego jest realizacja projektu w grupie. Do tej pory realizowane przez nas projekty były samodzielne. W tej sytuacji przy pomocy systemu kontroli wersji GIT mogliśmy rozwijać osobne funkcjonalności witryny nie ingerując w pracę kolegi z zespołu. Wspólnie wnioskujemy, że nauczyliśmy się nie tylko podstawowych elementów tworzenia tego typu witryn, ale przede wszystkim pracy w grupie – co będzie dla nas pomocne w realizacji przyszłych „większych projektów” na studiach oraz w pracy.

Uważamy, że poświęcony bardzo długi czas przeznaczony na realizację projektu dał nam nowe doświadczenia, a efekt końcowy jest dla nas zadowalający.

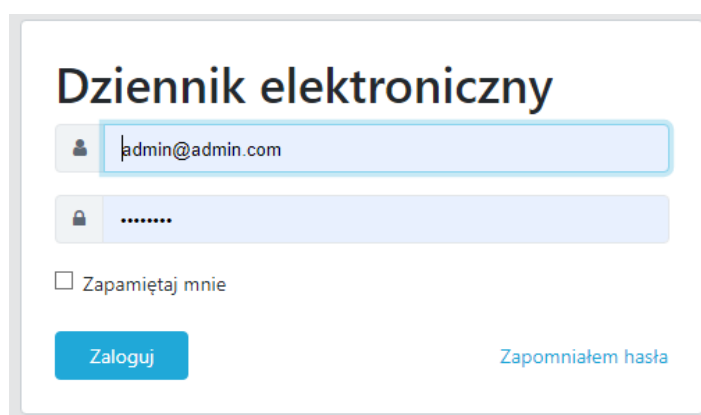
## Link do repozytorium

Nasz projekt wraz z dokumentacją oraz bazą danych został umieszczony na GitHubie można go znaleźć za pomocą poniższego linku

<https://github.com/Piotr2000PWSZ/Dziennik>

## Przykładowe zrzuty ekranu z aplikacji

Poniższe zdjęcie przedstawia panel logowania dla kont użytkowników

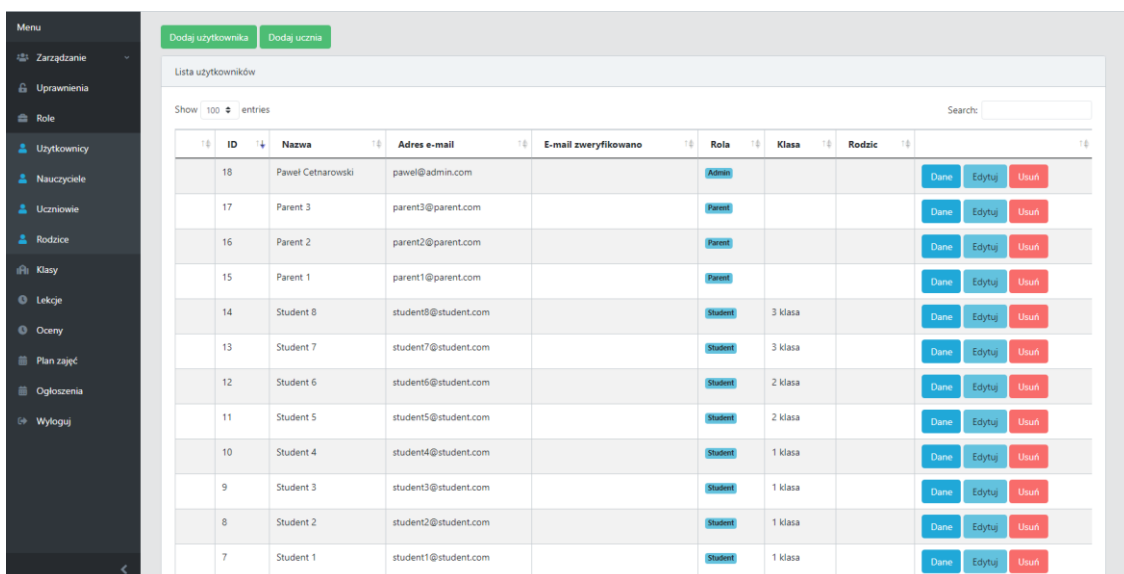


**Dziennik elektroniczny**

☐ Zapamiętaj mnie

Zaloguj [Zapomniałem hasła](#)

Poniższe zdjęcie przedstawia panel tworzenia i edycji użytkowników



Menu

- Zarządzanie
- Uprawnienia
- Role
- Użytkownicy
- Nauczyciele
- Uczniowie
- Rodzice
- Klasy
- Lekcje
- Oceny
- Plan zajęć
- Ogłoszenia
- Wyloguj

Dodaj użytkownika Dodaj ucznia

Lista użytkowników

Show 100 entries Search:

ID	Nazwa	Adres e-mail	E-mail zweryfikowano	Rola	Klasa	Rodzic	
18	Paweł Cetnarowski	pawel@admin.com		Admin			<button>Delete</button> <button>Edit</button> <button>Add</button>
17	Parent 3	parent3@parent.com		Parent			<button>Delete</button> <button>Edit</button> <button>Add</button>
16	Parent 2	parent2@parent.com		Parent			<button>Delete</button> <button>Edit</button> <button>Add</button>
15	Parent 1	parent1@parent.com		Parent			<button>Delete</button> <button>Edit</button> <button>Add</button>
14	Student 8	student8@student.com		Student	3 klasa		<button>Delete</button> <button>Edit</button> <button>Add</button>
13	Student 7	student7@student.com		Student	3 klasa		<button>Delete</button> <button>Edit</button> <button>Add</button>
12	Student 6	student6@student.com		Student	2 klasa		<button>Delete</button> <button>Edit</button> <button>Add</button>
11	Student 5	student5@student.com		Student	2 klasa		<button>Delete</button> <button>Edit</button> <button>Add</button>
10	Student 4	student4@student.com		Student	1 klasa		<button>Delete</button> <button>Edit</button> <button>Add</button>
9	Student 3	student3@student.com		Student	1 klasa		<button>Delete</button> <button>Edit</button> <button>Add</button>
8	Student 2	student2@student.com		Student	1 klasa		<button>Delete</button> <button>Edit</button> <button>Add</button>
7	Student 1	student1@student.com		Student	1 klasa		<button>Delete</button> <button>Edit</button> <button>Add</button>

Poniższe zdjęcie przedstawia panel do tworzenia oraz edycji uprawnień

**Dziennik**

Menu

- Zarządzanie
- Uprawnienia**
- Role
- Użytkownicy
- Nauczyciele
- Uczniowie
- Rodzice
- Klasy
- Lekcje
- Oceny
- Plan zajęć
- Ogłoszenia
- Wyloguj

**Dodaj nowe uprawnienie**

Lista uprawnień użytkowników

Show 100 entries

ID	Nazwa	Widok	Edytuj	Usuń
27	announcement_create	Widok	Edytuj	Usuń
26	school_class_access	Widok	Edytuj	Usuń
25	school_class_delete	Widok	Edytuj	Usuń
24	school_class_show	Widok	Edytuj	Usuń
23	school_class_edit	Widok	Edytuj	Usuń
22	school_class_create	Widok	Edytuj	Usuń
21	lesson_access	Widok	Edytuj	Usuń
20	lesson_delete	Widok	Edytuj	Usuń

Poniższe zdjęcie przedstawia wszystkie klasy jakie są dostępne w bazie z tego miejsca jest możliwość podejrzenia harmonogramu klasy oraz dodanie i edycja klasy

**Dziennik**

Menu

- Zarządzanie
- Klasy**
- Lekcje
- Oceny
- Plan zajęć
- Ogłoszenia
- Wyloguj

**Dodaj klasę**

Istniejące klasy

Show 100 entries

ID	Klasa	Wychowawca	Harmonogram	Zobacz	Edytuj	Usuń
4	4 klasa		Zobacz harmonogram	Zobacz	Edytuj	Usuń
3	3 klasa		Zobacz harmonogram	Zobacz	Edytuj	Usuń
2	2 klasa		Zobacz harmonogram	Zobacz	Edytuj	Usuń
1	1 klasa		Zobacz harmonogram	Zobacz	Edytuj	Usuń

Showing 1 to 4 of 4 entries

Previous 1 Next

Poniższe zdjęcie kreator tworzenia lekcji w tym miejscu można tworzyć edytować i lekcje oraz sprawdzać obecność

**Dziennik**

Menu

- Zarządzanie
- Klasy
- Lekcje**
- Oceny
- Plan zajęć
- Ogłoszenia
- Wyloguj

**Utwórz lekcję**

Informacje o lekcjach

Show 100 entries

ID	Klasa	Nauczyciel	Dzień tygodnia	Tydzień	Godzina rozpoczęcia	Godzina zakończenia	Obecność	Przedmiot	Zobacz	Edytuj	Usuń
17	4 klasa	Teacher 3	5	4	08:00	09:00	Lista obecności	WF	Zobacz	Edytuj	Usuń
16	1 klasa	Teacher 5	5	1	12:00	14:00	Lista obecności	Technika	Zobacz	Edytuj	Usuń
15	1 klasa	Teacher 1	5	1	10:00	12:00	Lista obecności	Geografia	Zobacz	Edytuj	Usuń
14	1 klasa	Teacher 2	5	1	08:00	10:00	Lista obecności	Zajęcia wychowawcze	Zobacz	Edytuj	Usuń
13	1 klasa	Teacher 3	4	1	14:00	16:00	Lista obecności	Religia	Zobacz	Edytuj	Usuń
12	1 klasa	Teacher 2	4	1	12:00	14:00	Lista obecności	Informatyka	Zobacz	Edytuj	Usuń
11	1 klasa	Teacher 1	4	1	10:00	12:00	Lista obecności	Religia	Zobacz	Edytuj	Usuń



Poniższe zdjęcie przedstawia przykładowy plan zajęć

Dziennik

Menu	Time	Poniedziałek	Wtorek	Środa	Czwartek	Piątek	Sobota	Niedziela
Zarządzanie	08:00 - 08:30							
Klasy	08:30 - 09:00		1 klasa Nauczyciel: Teacher 2 Przedmiot: Biologia			1 klasa Nauczyciel: Teacher 2 Przedmiot: Zajęcia wychowawcze		
Lekcje	09:00 - 09:30							
Oceny	09:30 - 10:00							
Plan zajęć	10:00 - 10:30							
Ogłoszenia	10:30 - 11:00	1 klasa Nauczyciel: Teacher 4 Przedmiot: Język polski	1 klasa Nauczyciel: Teacher 4 Przedmiot: Muzyka	1 klasa Nauczyciel: Teacher 5 Przedmiot: Plastyka	1 klasa Nauczyciel: Teacher 1 Przedmiot: Religia	1 klasa Nauczyciel: Teacher 1 Przedmiot: Geografia		
Wyloguj	11:00 - 11:30							
	11:30 - 12:00							
	12:00 - 12:30							
	12:30 - 13:00	1 klasa Nauczyciel: Teacher 5 Przedmiot: Matematyka	1 klasa Nauczyciel: Teacher 3 Przedmiot: Technika	1 klasa Nauczyciel: Teacher 1 Przedmiot: WF	1 klasa Nauczyciel: Teacher 2 Przedmiot: Informatyka	1 klasa Nauczyciel: Teacher 5 Przedmiot: Technika		
	13:00 - 13:30							
	13:30 - 14:00							
	14:00 - 14:30							
	14:30 - 15:00	1 klasa Nauczyciel: Teacher 3 Przedmiot: Język angielski		1 klasa Nauczyciel: Teacher 2 Przedmiot: Informatyka	1 klasa Nauczyciel: Teacher 3 Przedmiot: Religia			
	15:00 - 15:30							
	15:30 - 16:00							

Poniższe zdjęcie przedstawia ogłoszenia

Dziennik

Menu	Dodaj ogłoszenie			
Zarządzanie	Ogłoszenia			
Klasy	Id	Temat	Opis	Data utworzenia
Lekcje	1	Prośba	W szkole obowiązuje zmiana obuwia!	2022-06-10 17:20:55
Oceny	2	Egzamin	Egzamin musi się odbyć!	2022-06-11 15:43:07
Plan zajęć	3	Zajęcia odwołane	W dniu 24.05.2022 zajęcia z matematyki są odwołane dla klasy 2.	2022-06-11 16:19:29
Ogłoszenia				
Wyloguj				

Poniższe zdjęcie przedstawia logowanie na konto studenta

## Dziennik elektroniczny

☐ Zapamiętaj mnie

Zaloguj

Zapomniałem hasła

Poniższe zdjęcie przedstawia panel studenta

Dziennik

Menu

Oceny

Plan zajęć

Ogłoszenia

Wyloguj

Calendar

Time	Poniedziałek	Wtorek	Środa	Czwartek	Piątek	Sobota	Niedziela
08:00 - 08:30							
08:30 - 09:00		1 klasa Nauczyciel: Teacher 2 Przedmiot: Biologia			1 klasa Nauczyciel: Teacher 2 Przedmiot: Zajęcia wychowawcze		
09:00 - 09:30							
09:30 - 10:00							
10:00 - 10:30							
10:30 - 11:00	1 klasa Nauczyciel: Teacher 4 Przedmiot: Język polski	1 klasa Nauczyciel: Teacher 4 Przedmiot: Muzyka	1 klasa Nauczyciel: Teacher 5 Przedmiot: Plastyka	1 klasa Nauczyciel: Teacher 1 Przedmiot: Religia	1 klasa Nauczyciel: Teacher 1 Przedmiot: Geografia		
11:00 - 11:30							
11:30 - 12:00							
12:00 - 12:30							
12:30 - 13:00	1 klasa Nauczyciel: Teacher 5 Przedmiot: Matematyka	1 klasa Nauczyciel: Teacher 3 Przedmiot: Technika	1 klasa Nauczyciel: Teacher 1 Przedmiot: WF	1 klasa Nauczyciel: Teacher 2 Przedmiot: Informatyka	1 klasa Nauczyciel: Teacher 5 Przedmiot: Technika		
13:00 - 13:30							
13:30 - 14:00							
14:00 - 14:30							
14:30 - 15:00	1 klasa Nauczyciel: Teacher 3 Przedmiot: Język angielski		1 klasa Nauczyciel: Teacher 2 Przedmiot: Informatyka	1 klasa Nauczyciel: Teacher 3 Przedmiot: Religia			
15:00 - 15:30							
15:30 - 16:00							
16:00 - 16:30							