

POLITECHNIKA WROCŁAWSKA

WYDZIAŁ ELEKTRONIKI

## PROJEKT Z BAZ DANYCH

### Dziennik szkolny

Termin zajęć: Środa, 9-11

*Autorzy:*

Adam MIELNICZEK

Michał KOWALSKI

Maciej PAJĄK

*Prowadzący zajęcia:*

dr inż. Roman PTAK

Wrocław, 2020

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>3</b>
1.1	Cel projektu . . . . .	3
1.2	Zakres projektu . . . . .	3
<b>2</b>	<b>Opis wymagań</b>	<b>3</b>
2.1	Opis działania . . . . .	3
2.2	Rodzaje kont użytkownika . . . . .	3
2.3	Wymagania funkcjonalne . . . . .	4
2.3.1	Administrator . . . . .	4
2.3.2	Nauczyciel . . . . .	4
2.3.3	Wychowawca . . . . .	4
2.3.4	Uczeń . . . . .	5
2.3.5	Rodzic . . . . .	5
2.3.6	Kasa . . . . .	5
2.4	Wymagania niefunkcjonalne . . . . .	5
2.4.1	Wykorzystywane technologie i narzędzia . . . . .	5
2.4.2	Rozmiar bazy danych . . . . .	5
2.4.3	Bezpieczeństwo systemu bazodanowego . . . . .	5
<b>3</b>	<b>Projekt systemu</b>	<b>6</b>
3.1	Model konceptualny . . . . .	7
3.2	Model logiczny przed normalizacją . . . . .	8
3.3	Model fizyczny po normalizacji . . . . .	9
3.4	Sekwencje . . . . .	10
3.5	Widoki - przykład implementacji . . . . .	10
3.6	Indeksy . . . . .	11
3.7	Procedury składowe . . . . .	12
3.8	Wyzwalacze . . . . .	12
3.9	Przypadki użycia . . . . .	13
3.10	GUI i Struktura menu . . . . .	16
3.11	Bezpieczeństwo i łączenie z bazą danych . . . . .	19
<b>4</b>	<b>Implementacja i testy bazy danych</b>	<b>19</b>
4.1	Tworzenie tabel i definiowanie ograniczeń . . . . .	19
4.2	Implementacja uprawnień i innych zabezpieczeń . . . . .	20

4.3	Testowanie bazy danych na przykładowych danych . . . . .	20
<b>5</b>	<b>Implementacja i testy aplikacji</b>	<b>21</b>
5.1	Instalacja i konfigurowanie systemu . . . . .	21
5.2	Instrukcja użytkowania aplikacji . . . . .	23
5.3	Testowanie opracowanych funkcji systemu . . . . .	25
5.4	Omówienie wybranych rozwiązań programistycznych . . . . .	28
5.4.1	Interfejsu dostępu do bazy danych . . . . .	28
5.4.2	Implementacja mechanizmów bezpieczeństwa . . . . .	29
5.4.3	Implementacja funkcjonalności . . . . .	31
<b>6</b>	<b>Podsumowanie i wnioski</b>	<b>36</b>

# 1 Wstęp

## 1.1 Cel projektu

Celem projektu jest zapoznanie się z praktycznym użyciem (zaprojektowaniem oraz implementacją) narzędzia, jakim jest baza danych, zaznajomienie się z językiem SQL oraz poznanie sposobów na mądre przechowywanie i przeszukiwanie dużych zbiorów danych. Drugi cel to opracowanie aplikacji dostępowej, która będzie zapewniała dostęp do bazy danych osobom, nieposiadającym wiedzy z zakresu działania baz danych.

## 1.2 Zakres projektu

Projekt składać będzie się z 5 etapów:

1. Opis wymagań projektu,
2. Schemat konceptualny, logiczny oraz fizyczny relacji,
3. Implementacja bazy danych,
4. Napisanie aplikacji dostępowej dla użytkowników,
5. Prezentacja, testy i podsumowanie całości projektu.

# 2 Opis wymagań

## 2.1 Opis działania

Dziennik szkolny będzie zestawem bazy danych oraz aplikacji dostępowej, która umożliwi podstawowe zarządzanie szkołą. Docelowym klientem jest szkoła licealna średniej wielkości (około 400-500 uczniów). System ma spełniać podstawowe wymagania związane z funkcjonowaniem systemu nauczania w szkole takie jak np. ocenianie i sprawdzanie obecności uczniów na zajęciach.

## 2.2 Rodzaje kont użytkownika

Docelowy projekt będzie pozwalał na zalogowanie się na wymienione niżej rodzaje kont:

- Dyrektor (Administrator),
- Nauczyciel,

- Rodzic,
- Wychowawca,
- Uczeń,
- Kasa szkolna.

## 2.3 Wymagania funkcjonalne

Funkcjonalności są kategorii CRUD, chyba że w nawiasie w danym punkcie podano inaczej.

### 2.3.1 Administrator

- Zarządzanie uczniami,
- Zarządzanie nauczycielami,
- Zarządzanie klasami,
- Zarządzanie kontami użytkowników,
- Wprowadzanie nagan dyrektorskich uczniom,
- Generowanie świadectwa dla ucznia na zakończenie roku szkolnego - eksport do zewnętrznego pliku (C).

### 2.3.2 Nauczyciel

- Wpisywanie tematu zajęć i sprawdzanie listy obecności,
- Podgląd obecności danego ucznia z danej klasy (R),
- Podgląd oraz wpisywanie uczniom ocen cząstkowych i ich wag,
- Wystawianie ocen semestralnych i końcowych (także możliwość ich edycji lub usunięcia w razie przypadkowego wystawienia),
- Dodawanie terminów kartkówek, sprawdzianów itp. dla danej klasy,
- Wpisywanie uczniom uwag.

### 2.3.3 Wychowawca

- Dostęp do terminarza klasy wychowawczej - wychowawca ma możliwość podglądu wszystkich form (sprawdzianów, kartkówek) zaplanowanych w terminarzu danej klasy (R),
- Usprawiedliwianie nieobecności uczniów (U),
- Wystawianie ocen z zachowania, a także możliwość ich korekty lub usunięcia przy przypadkowym wpisaniu,

- Dostęp do statystyk każdego ucznia z klasy wychowawczej oraz całej klasy wychowawczej (R).

#### **2.3.4 Uczeń**

- Przegląd otrzymanych ocen i obecności (R),
- Przegląd terminarza (R).

#### **2.3.5 Rodzic**

- Przegląd otrzymanych ocen swojego dziecka (R),
- Przegląd dziennika uwag swojego dziecka (R),
- Przegląd nierozliczonych płatności (R),
- Usprawiedliwianie nieobecności swojego dziecka.

#### **2.3.6 Kasa**

- Dodawanie nowej płatności do ucznia lub całej klasy,
- Zatwierdzanie płatności w systemie,

### **2.4 Wymagania niefunkcjonalne**

#### **2.4.1 Wykorzystywane technologie i narzędzia**

- Dostęp z poziomu systemów Linuxowych i Windowsa (możliwa aplikacja webowa),
- Interfejs graficzny,
- Baza danych zrobiona w systemie MariaDB.

#### **2.4.2 Rozmiar bazy danych**

- Baza danych powinna pozwalać na założenie około 400-500 kont Uczniów, 2 razy większą ilość kont rodziców oraz około 30 kont nauczycieli,
- Przewidujemy, że w każdym roku szkolnym będzie przechowywane około 60 – 120 tys. ocen, około miliona obecności oraz około 30 tys. tematów lekcyjnych.
- Prawdopodobnie w "godzinach szczytu" z dziennika korzystać będzie około 80 osób.

#### **2.4.3 Bezpieczeństwo systemu bazodanowego**

- Logowanie się do systemu nazwą użytkownika i hasłem (hasło w bazie w postaci zaszyfrowanej),

- Możliwość zresetowania hasła na e-mail,
- Możliwość włączenia dwustopniowej weryfikacji (kod przysyłany na e-mail).

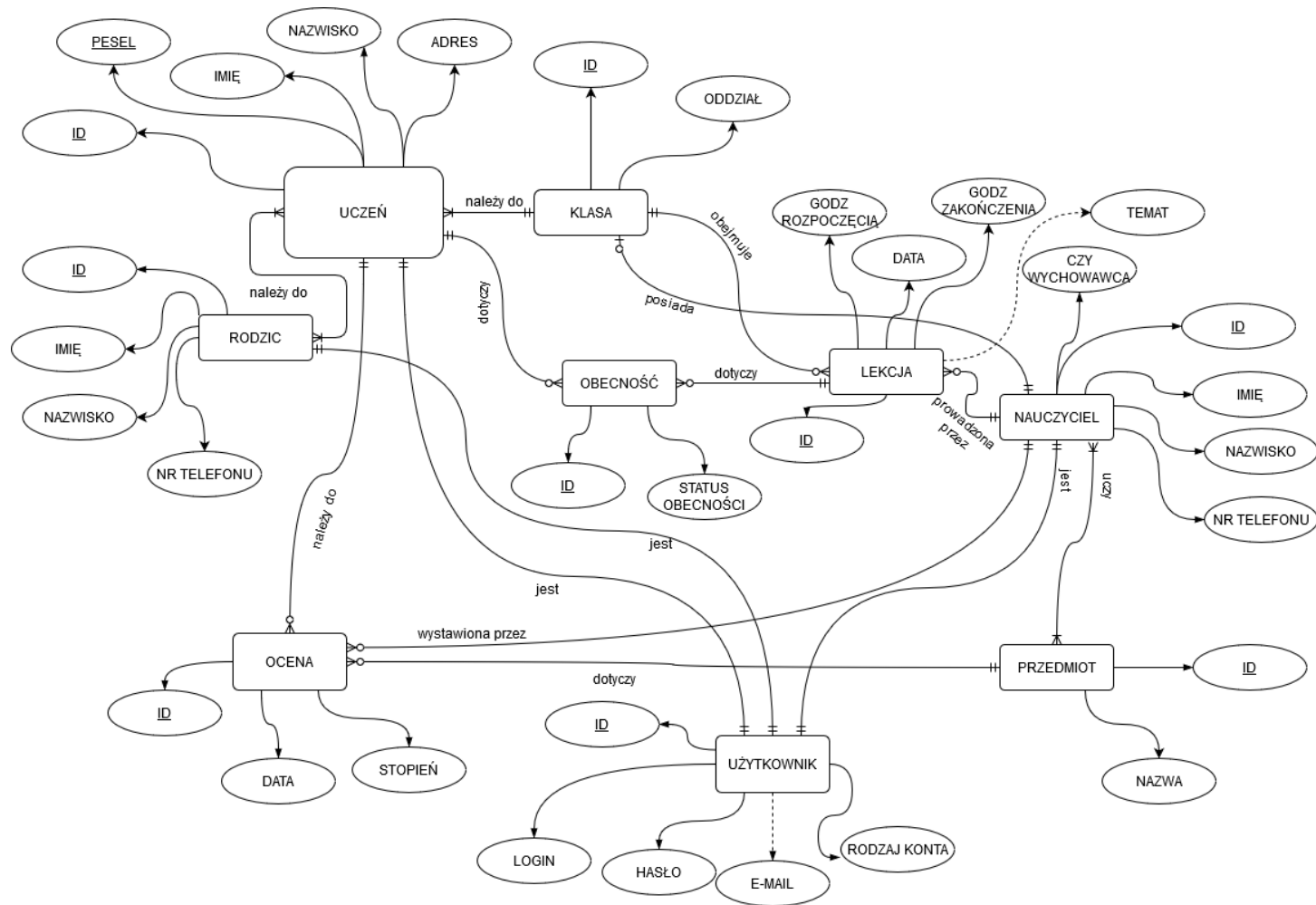
### 3 Projekt systemu

Poniżej przedstawione schematy systemu bazodanowego zostały zrobione w aplikacji webowej draw.io.

Przykładem normalizacji przeprowadzonym na poniższych modelach jest przeniesienie pól imię oraz nazwisko z relacji rodzic, uczeń i nauczyciel do relacji użytkownik. Adres został także rozdzielony na części oraz miejscowość i ulica przeniesione do dodatkowej relacji (podejrzewamy, że większość uczniów będzie z jednej miejscowości, a spora część także będzie mieszkać przy tej samej ulicy). Relacja nauczyciel przedmiotu dodana została w celu pozbycia się relacji wiele do wielu między nauczycielem, a nauczonym przez niego przedmiotem. To samo dotyczy relacji opiekunowie, która ułatwia połączenie rodzica z uczniem.

Ze względów bezpieczeństwa hasło przechowywane w bazie danych będzie szyfrowane algorytmem MD5 - stąd długość zaszyfrowanego hasła - zawsze 32 znaki.

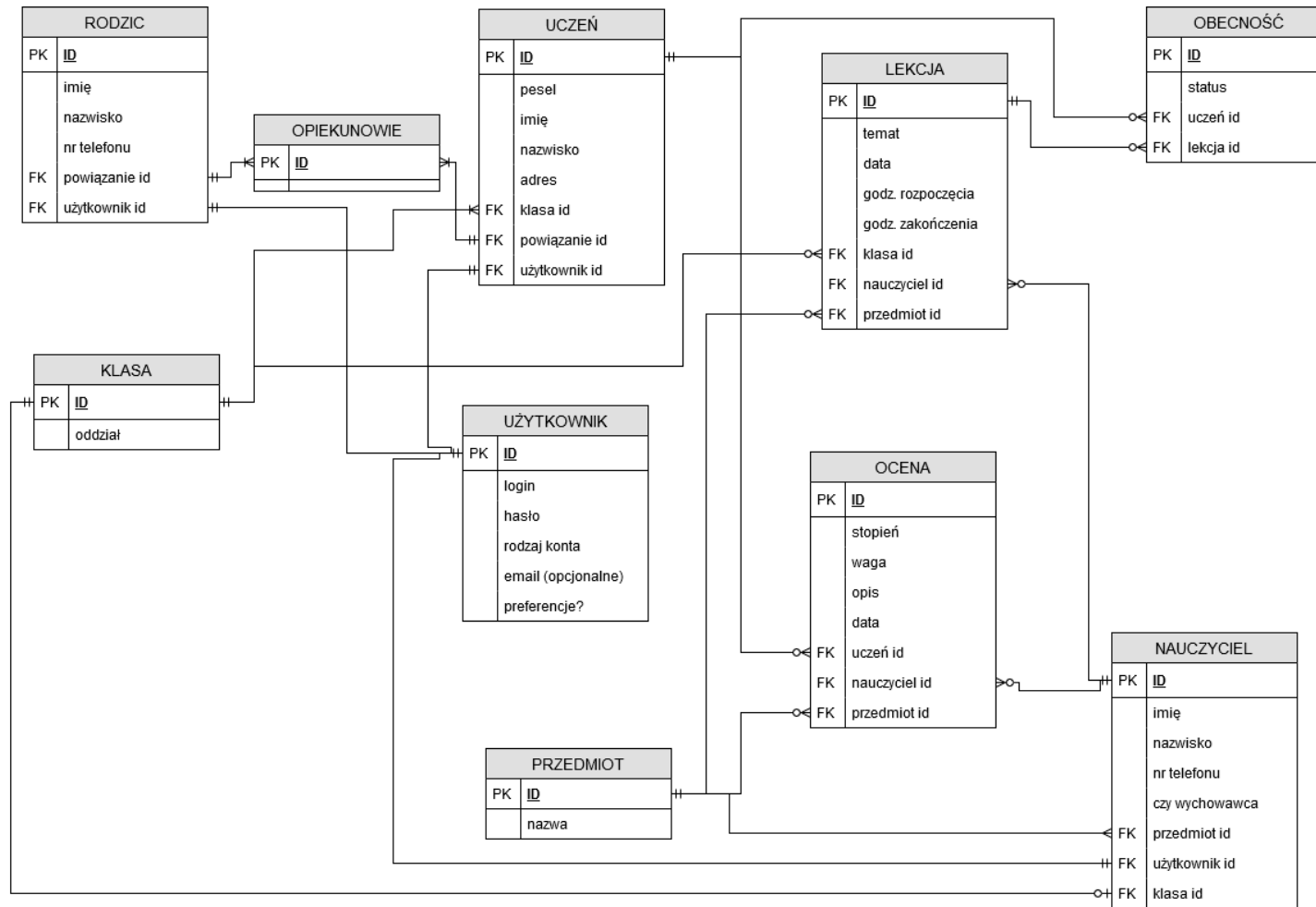
### 3.1 Model konceptualny



Rysunek 1: Model konceptualny encji dziennika elektronicznego

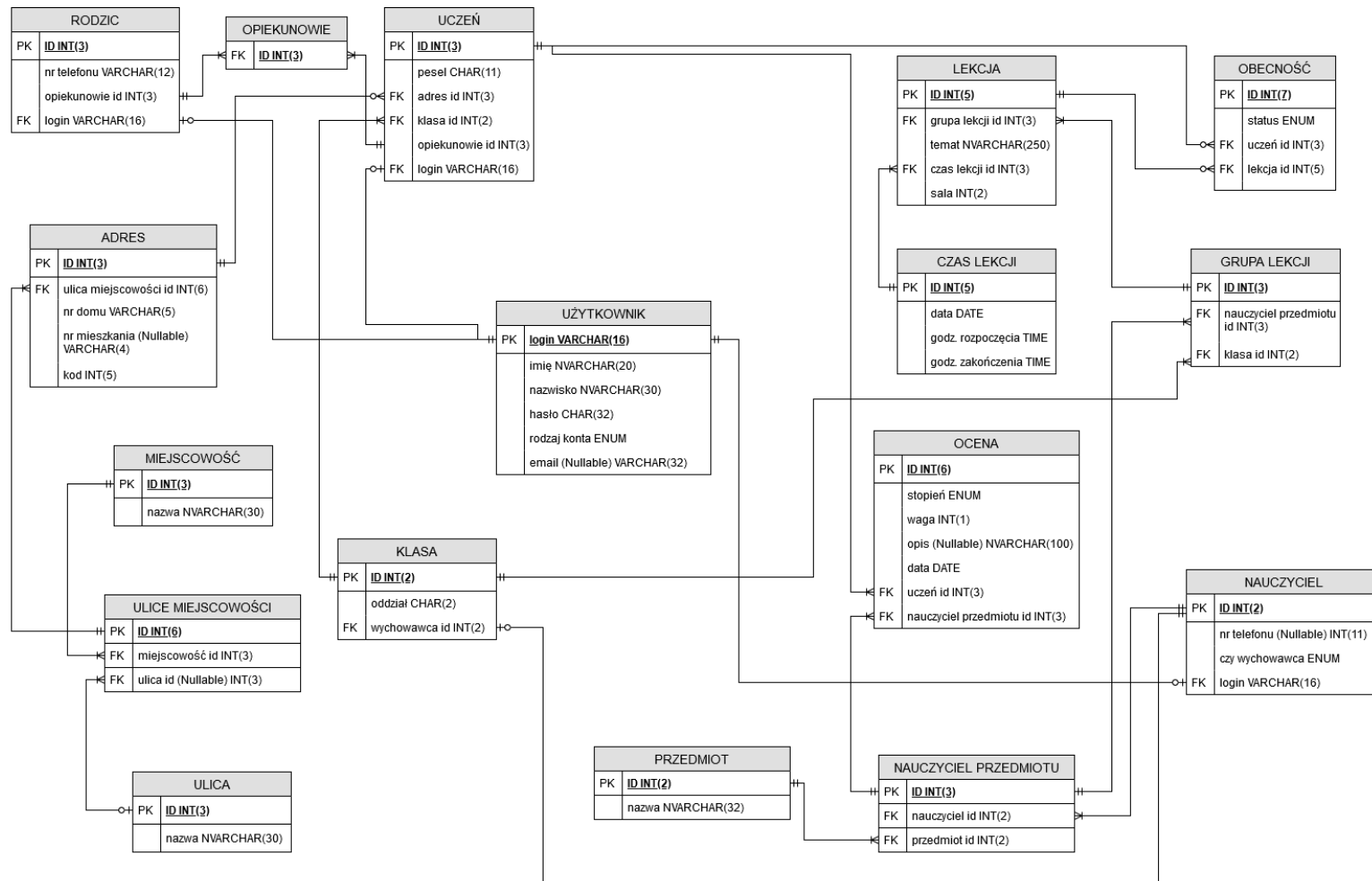


### 3.2 Model logiczny przed normalizacją



Rysunek 2: Model logiczny relacji dziennika elektronicznego

### 3.3 Model fizyczny po normalizacji



Rysunek 3: Model znormalizowany relacji dziennika elektronicznego

### 3.4 Sekwencje

Sekwencją jest obiekt w bazie danych generujący automatycznie liczby. Na przykład sztuczne klucze podstawowe. Przykładem sekwencji jest Autoincrement, który zwiększa wartość klucza podstawowego o 1 dla każdego nowego rekordu. Autoincrement został wykorzystany do generowania id w każdej relacji poza `uzytkownik` i `przedmiot` (ponieważ w tych dwóch relacjach klucze są naturalne).

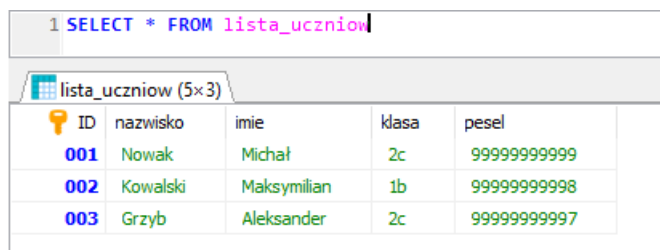
### 3.5 Widoki - przykład implementacji

Widok to obiekt, którego zadaniem jest wybrać odpowiednie dane z odpowiednich tabel i stworzyć z nich listę, do której odwoływać się można, jak do zwykłej tabeli.

Poniższy przykład pozwala na wyświetlenie listy uczniów.

```
1 CREATE VIEW lista_uczniow as
2 select `uczen`.`ID` AS `ID`,
3 `uzytkownik`.`nazwisko` AS `nazwisko`,
4 `uzytkownik`.`imie` AS `imie`,
5 `klasa`.`oddzial` AS `klasa`,
6 `uczen`.`pesel` AS `pesel`
7 from ((`uczen`
8 inner join `uzytkownik` on(`uczen`.`uzytkownik_login` = `uzytkownik`.`login`))
9 left join `klasa` on(`uczen`.`klasa_id` = `klasa`.`ID`))
```

Rysunek 4: Kod SQL widoku



ID	nazwisko	imie	klasa	pesel
001	Nowak	Michał	2c	99999999999
002	Kowalski	Maksymilian	1b	99999999998
003	Grzyb	Aleksander	2c	99999999997

Rysunek 5: Wywołanie widoku i wynik operacji

Poniżej przedstawiony został kod SQL kolejnych planowanych widoków:

```
1 CREATE VIEW adres_info_lista AS
2 SELECT `adres`.`adres_ID` AS `adres_ID`,
3 `miejscowosc`.`nazwa` AS `miejscowosc`,
4 `um`.`kod` AS `kod`,
5 `ulica`.`nazwa` AS `ulica`,
6 `adres`.`nr_domu` AS `nr_domu`,
7 `adres`.`nr_mieszkania` AS `nr_mieszkania`
8 FROM (((`ulice_miejscowosci` `um`
9 JOIN `ulica` ON(`um`.`ulica_id` = `ulica`.`ulica_ID`))
10 JOIN `miejscowosc` ON(`um`.`miejscowosc_id` = `miejscowosc`.`miejscowosc_ID`))
11 JOIN `adres` ON(`adres`.`ulice_miejscowosci_id` = `um`.`ulice_miejscowosci_ID`))
```

Rysunek 6: Widok tworzący listę adresów

```

1 CREATE VIEW lekcja_info_lista AS
2 SELECT `l`.`lekcja_ID` AS `lekcja_ID`,
3 `c`.`data` AS `data`,
4 `c`.`godz_start` AS `godz_start`,
5 `c`.`godz_koniec` AS `godz_koniec`,
6 `o`.`oddzial` AS `oddzial`,
7 `l`.`sala` AS `sala`,
8 `o`.`przedmiot` AS `przedmiot`,
9 `o`.`imie` AS `imie`,
10 `o`.`nazwisko` AS `nazwisko`,
11 `l`.`temat` AS `temat`,
12 `o`.`ID` AS `nauczyciel_id`
13 FROM ((`dziennik szkolny`.`lekcja` `l`
14 JOIN `dziennik szkolny`.`czas_lekcji` `c` ON(`l`.`czas_lekcji_id` = `c`.`czas_lekcji_ID`))
15 JOIN
16 (
17 SELECT `gr`.`grupa_lekcji_ID` AS `grupa_lekcji_ID`,
18 `k`.`oddzial` AS `oddzial`,
19 `n1st`.`imie` AS `imie`,
20 `n1st`.`nazwisko` AS `nazwisko`,
21 `n1st`.`przedmiot` AS `przedmiot`,
22 `n1st`.`ID` AS `ID`
23 FROM ((`dziennik szkolny`.`grupa_lekcji` `gr`
24 JOIN `dziennik szkolny`.`nauczyciel_info_lista` `n1st` ON(`gr`.`nauczyciel_przedmiot_id` = `n1st`.`nauczyciel_przedmiotu_ID`))
25 JOIN `dziennik szkolny`.`klasa` `k` ON(`gr`.`klasa_id` = `k`.`klasa_ID`))
26 )
27 `o` ON(`l`.`grupa_lekcji_id` = `o`.`grupa_lekcji_ID`))

```

Rysunek 7: Widok tworzący listę lekcji

```

1 CREATE VIEW nauczyciel_info_lista AS
2 SELECT `nauczyciel`.`nauczyciel_ID` AS `ID`,
3 `uzytkownik`.`nazwisko` AS `nazwisko`,
4 `uzytkownik`.`imie` AS `imie`,
5 `nauczyciel`.`nr_tel` AS `nr_tel`,
6 `uzytkownik`.`email` AS `email`,
7 `nauczyciel_przedmiotu`.`przedmiot_nazwa` AS `przedmiot`,
8 `nauczyciel_przedmiotu`.`nauczyciel_przedmiotu_ID` AS `nauczyciel_przedmiotu_ID`
9 FROM ((`nauczyciel`
10 JOIN `uzytkownik` ON(`nauczyciel`.`uzytkownik_login` = `uzytkownik`.`uzytkownik_login`))
11 JOIN `nauczyciel_przedmiotu` ON(`nauczyciel`.`nauczyciel_ID` = `nauczyciel_przedmiotu`.`nauczyciel_id`))

```

Rysunek 8: Widok tworzący listę nauczycieli

```

1 CREATE VIEW ocena_info_lista AS
2 SELECT `ocena`.`ocena_ID` AS `ocena_ID`,
3 `ocena`.`stopien` AS `stopien`,
4 `ocena`.`waga` AS `waga`,
5 `ocena`.`opis` AS `opis`,
6 `ocena`.`data` AS `data`,
7 `ocena`.`uczen_id` AS `uczen_id`,
8 `n`.`imie` AS `imie`,
9 `n`.`nazwisko` AS `nazwisko`,
10 `n`.`przedmiot` AS `przedmiot`
11 FROM (`ocena`
12 JOIN `nauczyciel_info_lista` `n` ON(`ocena`.`nauczyciel_przedmiotu_id` = `n`.`nauczyciel_przedmiotu_ID`))

```

Rysunek 9: Widok tworzący listę ocen

```

1 CREATE VIEW rodzic_info_lista AS
2 SELECT `r`.`rodzic_ID` AS `rodzic_ID`,
3 `r`.`nr_telefonu` AS `nr_telefonu`,
4 `u`.`imie` AS `imie`,
5 `u`.`nazwisko` AS `nazwisko`,
6 `u`.`email` AS `email`,
7 `r`.`opiekunowie_id` AS `opiekunowie_id`
8 FROM (`rodzic` `r`
9 LEFT JOIN `uzytkownik` `u` ON(`r`.`uzytkownik_login` = `u`.`uzytkownik_login`))

```

Rysunek 10: Widok tworzący listę rodziców

## 3.6 Indeksy

Indeksy służą do szybkiego odnajdywania danych. Szczególnie przydatne są, gdy z bardzo dużej tabeli potrzebny jest tylko jeden lub kilka rekordów. W dzienniku elek-

tronicznym indeksy będą wykorzystywane w relacjach: Ocena (ID), Lekcja (ID oraz czas lekcji ID) oraz Obecność (ID). Indeksy zostaną użyte także do daty lekcji, daty oceny, oraz indeks zgrupowany z imieniem i nazwiskiem (często te dane wyciągamy razem, oraz po nich sortujemy). Użyte zostaną indeksy w formie implementacji jaką jest drzewo binarne BTEE.

### 3.7 Procedury składowe

Procedury składowe pozwalają na wyciąganie potrzebnych danych z odpowiednich tabel. Zapewniają możliwość komunikacji aplikacji dostępowej z bazą danych.

Przykładem procedury jest wyświetlenie listy ocen danego ucznia.

```
1 CREATE DEFINER='root'@'localhost' PROCEDURE `lista_ocen_ucznia`(  
2   IN `in_uczen_id` INT  
3 )  
4  
5 LANGUAGE SQL  
6 NOT DETERMINISTIC  
7 CONTAINS SQL  
8 SQL SECURITY DEFINER  
9 COMMENT ''  
10 select ocena.uczen_id, ocena.stopien, ocena.waga, ocena.opis, ocena.`data`, nauczycielprzedmiotu.nauczyciel_id, nauczycielprzedmiotu.przedmiot_nazwa  
11 from ((ocena  
12 inner join uczen on ocena.uczen_id = uczen.ID and uczen.ID = in_uczen_id)  
13 left join nauczycielprzedmiotu on ocena.nauczyciel_przedmiot_id = nauczycielprzedmiotu.ID)
```

Rysunek 11: Kod SQL tworzący procedurę

1 CALL `lista\_ocen\_ucznia`('1')

Wynik #1 (7x3)

uczen_id	stopien	waga	opis	data	nauczyciel_id	przedmiot_nazwa
001	dop	3		2020-04-29	01	WOS
001	ndst	2		2020-04-29	02	matematyka
001	db	1		2020-04-29	02	matematyka

Rysunek 12: Wywołanie procedury i wynik operacji

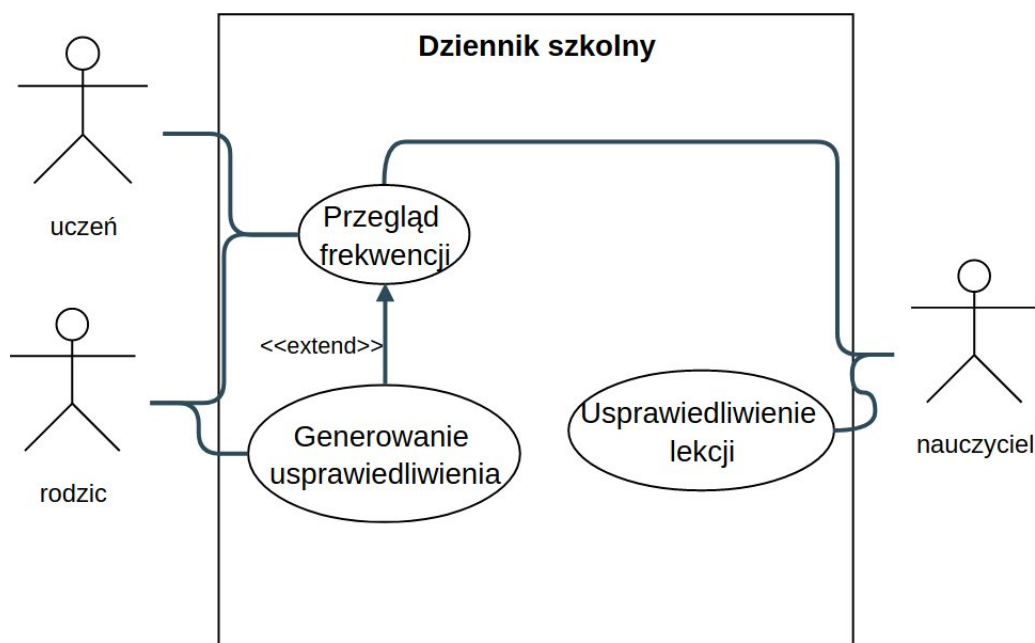
### 3.8 Wyzwalacze

Wyzwalacze (triggery) są to reakcje na pewne zdarzenia w bazie danych, takie jak np. usunięcie danego rekordu, zmiany jego wartości lub dodanie nowego. Na ten moment nie została przewidziana potrzeba stosowania ich w tym dzienniku.

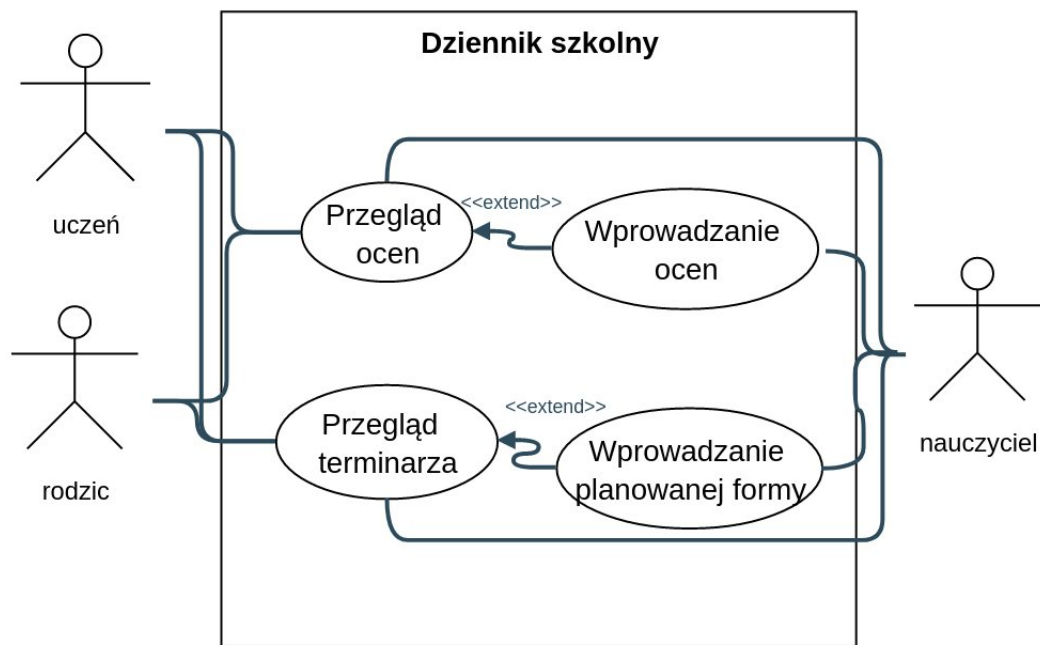
Dziennik szkolny oprócz bazy danych składać się będzie z aplikacji dostępowej, której menu po zalogowaniu na odpowiednie konto będzie się składało z zakładki zależnych od rodzaju konta użytkownika. Główną zakładką będzie zarządzanie kontem użytkownika - podgląd danych oraz zmiana hasła logowania. Diagramy i makiety zrobione przy pomocy `moqups.com`.

### 3.9 Przypadki użycia

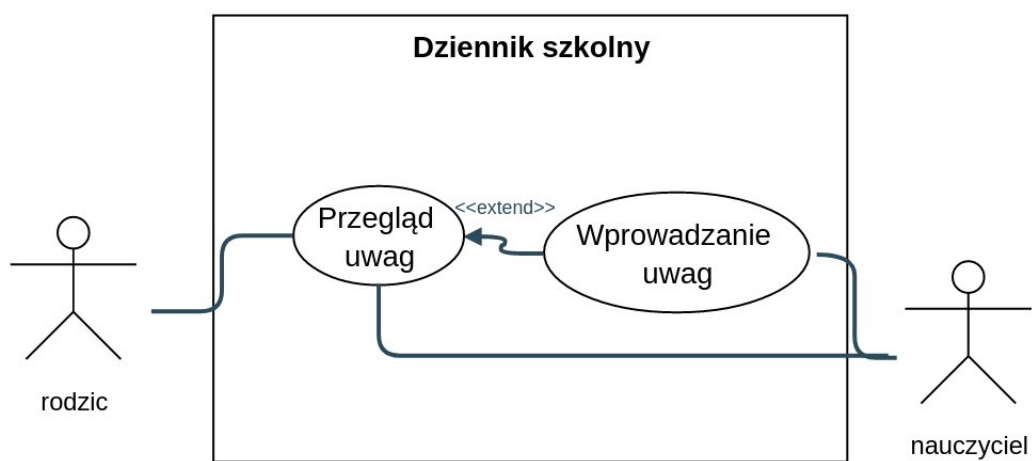
Poniższy diagram przedstawia trzy przypadki użycia dziennika szkolnego - uczeń lub rodzic zauważa nieobecność, rodzic wysyła usprawiedliwienie, a nauczyciel je zatwierdza.



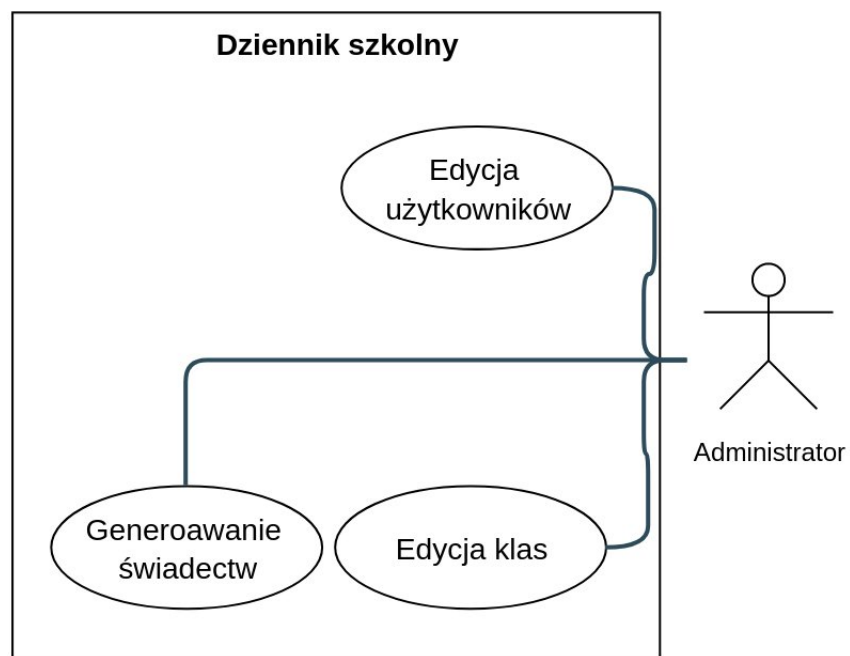
Rysunek 13: Diagram przypadków użycia związanych z usprawiedliwianiem nieobecności (edycją frekwencji)



Rysunek 14: Diagram przypadków użycia związanych z ocenianiem i terminarzem



Rysunek 15: Diagram przypadków użycia związanych z wystawianiem i odczytem uwag



Rysunek 16: Diagram przypadków użycia związanych z administratorem systemu bazodanowego



### 3.10 GUI i Struktura menu

The login screen features a large title 'Dziennik elektroniczny' at the top. Below it, there are two input fields: 'Login:' with a placeholder 'Login' and 'Hasło:' with a placeholder '\*\*\*\*\*'. A blue link 'Nie pamiętam hasła' is positioned below the password field. A 'Zaloguj' button is located to the right of the password field.

Rysunek 17: Ekran logowania

The admin dashboard is titled 'Zalogowano jako Admin'. It features a horizontal menu with seven tabs: 'Zarządzanie kontem', 'Edycja kont użytkownika', 'Edycja uczniów', 'Edycja rodziców', 'Edycja pracowników', 'Edycja klas', and 'Nagany'. The 'Zarządzanie kontem' tab is currently selected. Below the menu, the 'Dane:' section displays 'login: a123' and 'email: ABC@DEF.pl'. The 'Zmiana hasła:' section contains three password input fields labeled 'stare hasło:', 'nowe hasło:', and 'nowe hasło jeszcze raz:', each with a '\*\*\*\*\*' placeholder. A 'Zmiana' button is located at the bottom right.

Rysunek 18: Zakładki widoczne z poziomu konta administratora

# Zalogowano jako Nauczyciel

Zarządzanie kontem	Lekcje (temat i frekwencja)	Oceny	Oceny końcowe
Terminarz	Uwagi	Klasa wychowawcza	

**Dane:**

Imię: XYZ

Nazwisko: XYZ

Nr tel: 123987654

login: n123

email: ABC@DEF.pl

**Zmiana hasła:**

stare hasło:

nowe hasło:

nowe hasło jeszcze raz:

Rysunek 19: Zakładki widoczne z poziomu konta nauczyciela

# Zalogowano jako rodzic

Zarządzanie kontem	Oceny	Dziennik uwag	Płatność	Usprawiedliwieni
--------------------	-------	---------------	----------	------------------

**Dane:**

imię: ABCDEFG

nazwisko: ABCDEFG

nr tel: +48123456789

login: r123

email: ABC@DEF.pl

**Zmiana hasła:**

stare hasło:

nowe hasło:

nowe hasło jeszcze raz:

Rysunek 20: Zakładki widoczne z poziomu konta rodzica

# Zalogowano jako uczeń

Zarządzanie kontemOcenyFrekwencjaTerminarz form

Dane:  
imię: ABCDEFG  
nazwisko: ABCDEFG  
klasa: 1A  
pesel: 12345678901  
adres: ABC 15a/2, 01-234, Wrocław  
login: u123  
email: ABC@DEF.pl

Zmiana hasła:  
stare hasło:  
  
nowe hasło:  
  
nowe hasło jeszcze raz:  
  

Zmiana

Rysunek 21: Zakładki widoczne z poziomu konta ucznia

### 3.11 Bezpieczeństwo i łączenie z bazą danych

Aplikacja webowa napisana w PHP łączyć się będzie z bazą za pomocą `mysqli`. Głównym mechanizmem bezpieczeństwa, użytym w projekcie jest oczywiście konieczność wpisania poprawnego loginu oraz hasła, żeby móc czytać lub edytować dane osobowe. Hasło w bazie danych jest przechowywane w postaci zaszyfrowanej przez `md5`. Szyfrowanie odbywa się z poziomu aplikacji dostępowej i jest porównywane z zaszyfrowanym będącym w bazie. Do zmiany hasła konieczna jest znajomość starego, oraz dwukrotne powtórzenie nowego.

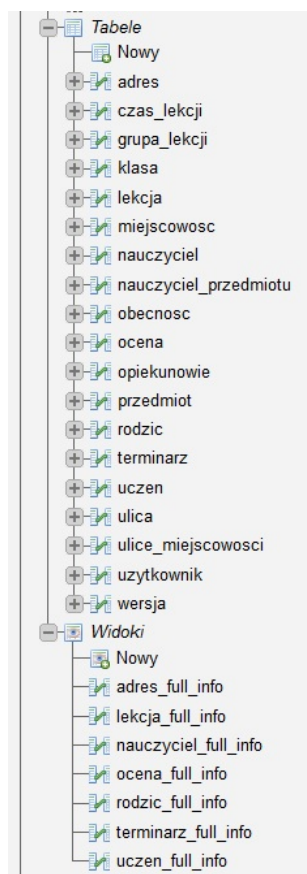
## 4 Implementacja i testy bazy danych

### 4.1 Tworzenie tabel i definiowanie ograniczeń

Implementacja została wykonana za pomocą wcześniej wybranego programu do zarządzania bazami danych – HeidiSQL. Serwer dla naszej bazy danych postawiony został lokalnie używając MariaDB. Następnie na podstawie modelu znormalizowanego dodaliśmy potrzebne tabele i ograniczenia.



Rysunek 22: Lista zaimplementowanych procedur i funkcji



Rysunek 23: Lista zaimplementowanych tabel i widoków

## 4.2 Implementacja uprawnień i innych zabezpieczeń

Do bazy danych został dodany użytkownik posiadający pełne prawa - będzie to przydatne w niektórych funkcjonalnościach administratora systemu i nauczyciela. Drugi użytkownik ma prawa do odczytu, jednak jego prawa do zapisu są bardzo ograniczone - to będzie dla ucznia i rodzica. Resztę zabezpieczeń związanych z uprawnieniami przejmie aplikacja dostępowa.

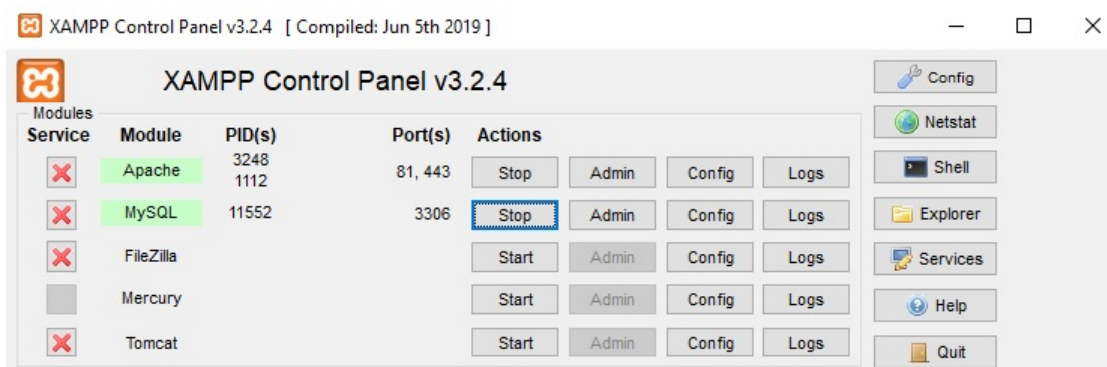
## 4.3 Testowanie bazy danych na przykładowych danych

Dla sprawdzenia poprawności działania bazy użyty został generator losowych danych zawarty w MS Excelu . Wygenerowane zostało 100 uczniów, 200 rodziców i 60 nauczycieli. Do tego zajęcia, oceny i frekwencja. Pozwoliło to na testowanie między innymi, czy widoki zwracają odpowiednie dane. Do tego wygenerowane dane będą mocno przydatne podczas implementacji i testowania aplikacji dostępowej.

## 5 Implementacja i testy aplikacji

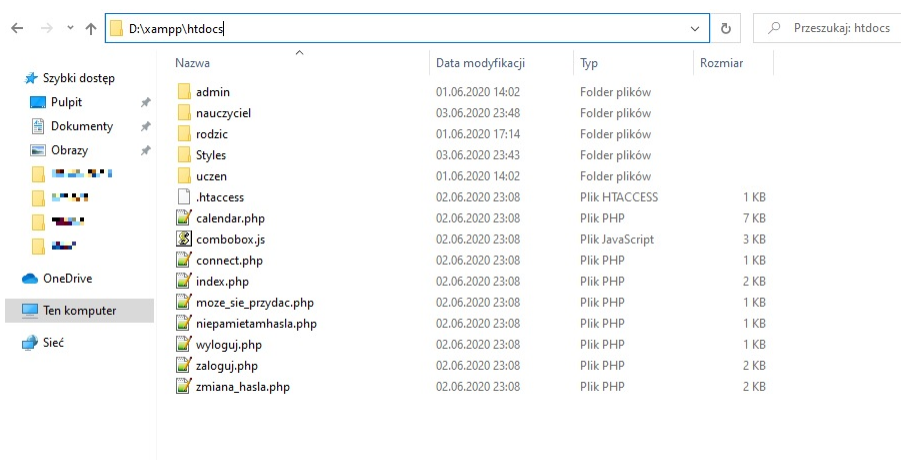
### 5.1 Instalacja i konfigurowanie systemu

Na komputerze klienta potrzebne będzie zainstalowanie serwera WWW rozumiejącego język PHP oraz hostującego bazę danych. W przypadku, gdy serwerem ma być komputer z systemem Linux - będzie potrzebny pakiet LAMP. Na Windowsie przyda się aplikacja taka jak XAMPP - można ją pobrać za darmo ze strony producenta. Program ten należy uruchomić jako administrator i włączyć Apache oraz MySQL.



Rysunek 24: Okno programu XAMPP po uruchomieniu Apache'a i MySQL'a

Następnie należy skopiować pliki aplikacji dostępowej do folderu serwera. Domyślnie jest to ~/public\_html/ na Linuxie lub folder\_xampp/htdocs/ na Windowsie.



Rysunek 25: Folder htdocs po wklejeniu plików

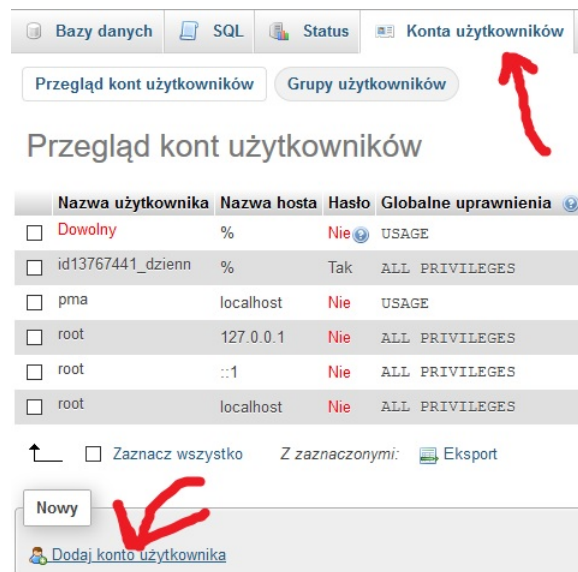
Kolejnym krokiem będzie kliknięcie na przycisk Admin przy MySQL, po otwarciu strony phpMyAdmin należy utworzyć nową bazę o nazwie dziennik.

Rysunek 26: Tworzenie bazy danych

Następnie konieczne będzie przejście do zakładki import i zaimportowanie pliku baza.sql.

Rysunek 27: Importowanie systemu bazodanowego

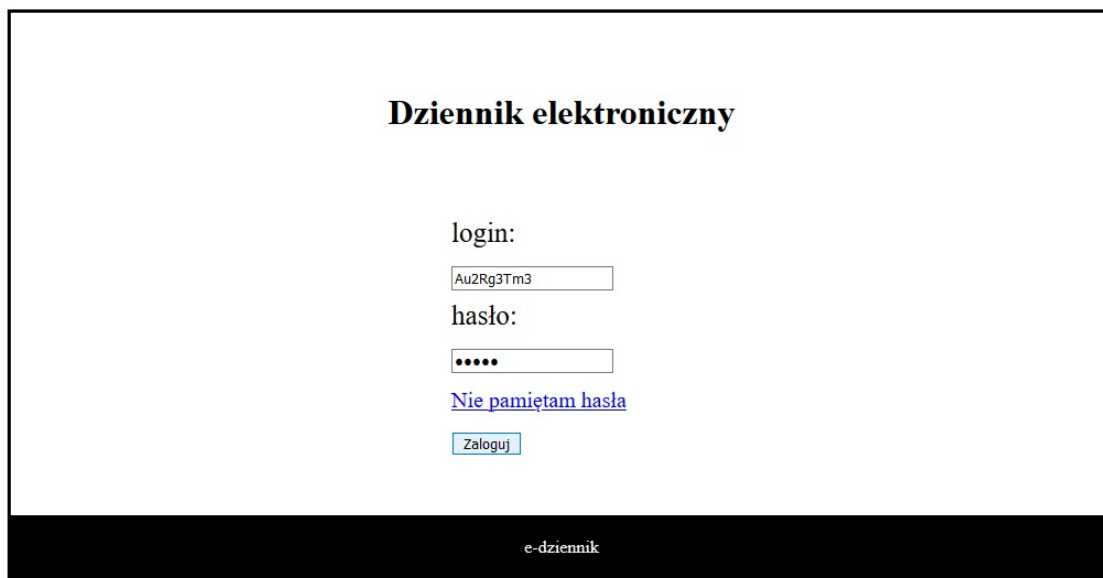
Ostatecznie w zakładce uprawnienia dodanie konta użytkownika bazy - z pełnymi uprawnieniami.



Rysunek 28: Dodawanie konta użytkownika

## 5.2 Instrukcja użytkowania aplikacji

Aby połączyć się z aplikacją należy w przeglądarce wpisać adres IP komputera, na którym ona działa. W naszym przypadku jest to po prostu localhost. Powinno otworzyć się okno logowania do systemu.



Rysunek 29: Ekran logowania

Po wpisaniu poprawnego loginu i hasła otworzy się strona z danymi osobowymi zalogowanego użytkownika. Używając zakładek można przełączać się między różnymi funkcjonalnościami dziennika.



**Zalogowano jako uczeń**

Zarządzanie kontem Oceny Frekwencja Terminarz

**Dane:**

imie: WERONIKA

nazwisko: KANIA

klasa: 3b

pesel: 23758491432

adres: Wojciecha Bogusławskiego 144/46, 14753 Niemce

login: Au2Rg3Tm3

email: przykladowy@mail.xd

**Zmiana hasła:**

stare hasło:

nowe hasło:

powtórz nowe hasło:

e-dziennik

Rysunek 30: Po zalogowaniu

Szczegóły niektórych elementów otwierają się po najechaniu na nie.

Zarządzanie kontem Oceny Frekwencja Terminarz			
<b>Oceny:</b>			
biologia	1	3	
chemia	0		
fizyka	6		
geografia	waga: 3 nauczyciel: PAULINA SOWA opis:		
język angielski			
język niemiecki	2		
język polski	3	3	
podstawy przedsiębiorczości	1		

Rysunek 31: Podgląd ocen

### Frekwencja:

2020-09-01	N	
2020-09-02	S	
2020-09-05	N	
2020-09-06	S	
2020-09-07	fizyka 09:45:00-10:30:00	
2020-09-08		
2020-09-10	S	S
2020-09-11	S	
2020-09-12	S	
2020-09-13	N	
2020-09-14	S	N

Rysunek 32: Podgląd frekwencji

## 5.3 Testowanie opracowanych funkcji systemu

Funkcjonalności aplikacji testowane były na bieżąco, podczas tworzenia. Dzięki bazie wypełnionej wygenerowanymi danymi było to możliwe. Dodatkowo część testów można było przeprowadzić, nie zaglądając do bazy danych.

Poniższy test przedstawia wpisanie oceny uczniowi, a potem jej podgląd z konta ucznia:

Nauczyciel po zalogowaniu wchodzi do zakładki oceny:

Zalogowano jako nauczyciel

Zarządzanie kontem
Lekcje
**Oceny**
Terminarz

**Oceny:**

Wybierz klasę

wyloguj

e-dziennik

Rysunek 33: Zakładka ocen z perspektywy nauczyciela

## Oceny:

---Wybierz klasę---

---Wybierz klasę---

1b wychowanie fizyczne

1d wychowanie fizyczne

3b wychowanie fizyczne

3c wychowanie fizyczne

4c wychowanie fizyczne

4d wychowanie fizyczne

Rysunek 34: Lista wyboru jednej z uczonych klas

Wpisanie oceny:

[illegible]

Rysunek 35: Oceny klasy i otwarte okno dodawania oceny

Sprawdzenie, czy ocena trafiła na listę ocen:

### Oceny:

---

Wybierz klasę---

▼

Nr	Imię i nazwisko	Oceny
1.	MARIA BEDNAREK	
2.	WERONIKA KANIA	4
3.	MICHALINA MAJ	
4.	ŁUKASZ MAJEWSKA	
5.	LEOKADIA SADOWSKI	
6.	HUBERT SAWICKI	
7.	WIKTORIA SZULC	
8.	CEZARY WIERZBICKI	
9.	WOJCIECH WILK	

Rysunek 36: Oceny klasy po dodaniu oceny

Zalogowanie się na konto ucznia:

**Zalogowano jako uczeń**

Zarządzanie kontemOcenyFrekwencjaTerminarz

**Dane:**  
imie: WERONIKA  
nazwisko: KANIA  
klasa: 3b

**Zmiana hasła:**  
stare hasło:  
  
nowe hasło:

Rysunek 37: Domyślna zakładka ucznia

Przejsie do zakładki oceny:

**Zalogowano jako uczeń**

Zarządzanie kontemOcenyFrekwencjaTerminarz

**Oceny:**

biologia	1	3
chemia	0	
fizyka	0	
geografia	3	
język angielski	5	
język niemiecki	2	
język polski	3	3
podstawy przedsiębiorczości	1	
wychowanie fizyczne	4	

wyloguj

e-dziennik

Rysunek 38: Zakładka oceny z perspektywy ucznia

Odczytanie oceny oraz jej szczegółów po najechaniu myszką:

#### Oceny:

biologia	1	3
chemia	0	
fizyka	0	
geografia	3	
język angielski	5	
język niemiecki	waga: 4 nauczyciel: AMELIA WOŹNIAK opis: sprawdzian	
język polski		
podstawy przedsiębiorczości		
wychowanie fizyczne	4	

Rysunek 39: Szczegóły oceny

<input type="checkbox"/>				173	0	2		2020-11-10	053	49	AMELIA	WOŹNIAK	wychowanie fizyczne
<input type="checkbox"/>				176	6	3		2020-09-03	023	49	AMELIA	WOŹNIAK	wychowanie fizyczne
<input type="checkbox"/>				196	2	8		2020-09-17	076	49	AMELIA	WOŹNIAK	wychowanie fizyczne
<input type="checkbox"/>				199	3	3		2020-11-23	095	49	AMELIA	WOŹNIAK	wychowanie fizyczne
<input type="checkbox"/>				260	6	1		2020-09-03	097	49	AMELIA	WOŹNIAK	wychowanie fizyczne
<input type="checkbox"/>				573	4	6	bieganie	2020-06-08	010	49	AMELIA	WOŹNIAK	wychowanie fizyczne
<input type="checkbox"/>				574	6	0	ctycuc ughxcet5yx6	2020-06-08	036	49	AMELIA	WOŹNIAK	wychowanie fizyczne
<input type="checkbox"/>				576	4	4	sprawdzian	2020-06-08	005	49	AMELIA	WOŹNIAK	wychowanie fizyczne
<input type="checkbox"/>				17	2	4		2020-10-06	082	50	KINGA	KOŹŁOWSKI	wychowanie fizyczne
<input type="checkbox"/>				31	2	8		2020-12-10	039	50	KINGA	KOŹŁOWSKI	wychowanie fizyczne
<input type="checkbox"/>				53	4	1		2020-09-24	072	50	KINGA	KOŹŁOWSKI	wychowanie fizyczne
<input type="checkbox"/>				82	5	2		2020-12-09	020	50	KINGA	KOŹŁOWSKI	wychowanie fizyczne
<input type="checkbox"/>				265	1	2		2020-09-11	046	50	KINGA	KOŹŁOWSKI	wychowanie fizyczne

Rysunek 40: Fragment widoku ocena\_full\_info ukazujący wpisaną ocenę

Powyższy test dowiódł poprawności działania jednego z przypadków użycia aplikacji dostępowej.

## 5.4 Omówienie wybranych rozwiązań programistycznych

Aplikacja dostępowa opiera się głównie na językach HTML oraz PHP. Dodatkowo niektóre funkcjonalności (np. wybór z listy) wymagały dodania skryptów wykonywanych po stronie użytkownika (javascript / jquery).

### 5.4.1 Interfejsu dostępu do bazy danych

Za dostęp aplikacji do bazy danych odpowiedzialny jest plik `connect.php` zawierający dane takie jak nazwa bazy, użytkownik, hasło oraz host. Dane te po pobraniu z tego pliku są używane za pomocą polecenia `mysqli`. Poniżej znajduje się przykład połączenia z bazą i pobierania z niej danych - w tym przypadku informacji osobowych ucznia.

```

<?php
unset($_SESSION['blad']);

require_once "../connect.php";

$conn=@new mysqli($IP, $username, $password, $DB_name);

if ($conn->connect_errno!=0)
{
    echo "Error: ".$conn->connect_errno;
}
else
{
    $login=$_SESSION['uzytkownik_login'];

    $haslo = $_SESSION['haslo'];

    $sql="SELECT * FROM uzytkownik WHERE uzytkownik_login='$login' AND haslo='$haslo'";
    $result = @$conn->query($sql);

    $sql2="SELECT * FROM uczen WHERE uzytkownik_login='$login' ";
    $result2 = @$conn->query($sql2);

    $dane_uzytkownika=@mysqli_fetch_assoc($result);
    $dane_ucznia=@mysqli_fetch_assoc($result2);

    $sql3="SELECT * FROM klasa WHERE klasa_ID='$dane_ucznia[klasa_id]' ";
    $result3 = @$conn->query($sql3);

    $klasa=@mysqli_fetch_assoc($result3);

    $sql4="SELECT * FROM adres_full_info WHERE adres_ID='$dane_ucznia[adres_id]' ";
    $result4 = @$conn->query($sql4);

    $adres=@mysqli_fetch_assoc($result4);

    $conn->close();
}
?>

```

Rysunek 41: Skrypt łączący się z bazą danych - przykład

#### 5.4.2 Implementacja mechanizmów bezpieczeństwa

Poza sprawdzaniem czy hasło podane przez użytkownika zgadza się z tym będącym w bazie, zaimplementowane zostały dwa ważne systemy bezpieczeństwa. Pierwszym z nich jest szyfrowanie hasła już w aplikacji. Baza danych jedynie dostaje szyfr MD5 i porównuje go z tym będącym w niej zapisanym. Powoduje to, że nawet administratorzy systemu bazodanowego nie będą w stanie poznać hasła jakiegoś użytkownika - wiele osób niestety używa tego samego hasła na różnych stronach internetowych.

```

<?php
    session_start();

    require_once "connect.php";

    $conn=new mysqli($IP, $username, $password, $DB_name);

    if ($conn->connect_errno!=0)
    {
        echo "Error: ".$conn->connect_errno;
    }
    else
    {
        $login = $_POST['login'];
        $haslo = $_POST['haslo'];
        $haslo_md5=md5($haslo);

        $sql="SELECT * FROM uzytkownik WHERE uzytkownik_login='$login' AND haslo='$haslo_md5'";

        if($result = @$conn->query($sql))
        {
            $czy_dobrze=$result->num_rows;

            if($czy_dobrze>0)
            {
                $dane=mysqli_fetch_assoc($result);
                $_SESSION['uzytkownik_login']=$dane['uzytkownik_login'];
                $_SESSION['haslo']=$dane['haslo'];
                $_SESSION['timeout']=180; // w sekundach

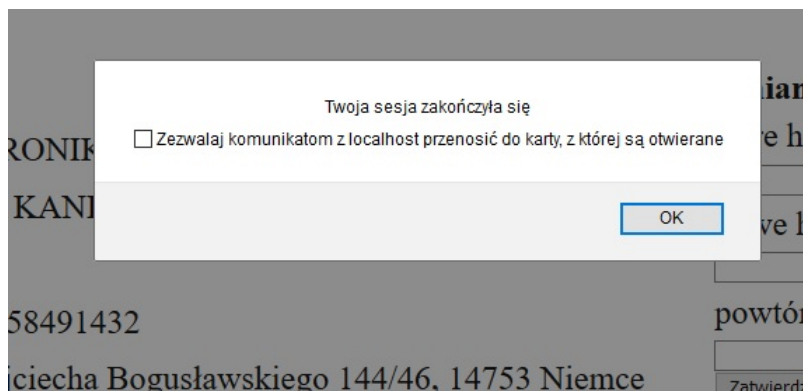
                $result->free_result();
                if($dane['rodzaj']=='U')
                {
                    header('Location: uczen/u_konto.php');
                }
                else if($dane['rodzaj']=='A')
                {
                    header('Location: admin/a_konto.php');
                }
                else if($dane['rodzaj']=='R')
                {
                    header('Location: rodzic/r_konto.php');
                }
                else if($dane['rodzaj']=='N')
                {
                    header('Location: nauczyciel/n_konto.php');
                }
                else
                {
                    $_SESSION['blad']='<span style="font-size:20px; color:red">Nieprawidłowy typ
konta</span>';
                    header('Location: index.php');
                }
            }
            else
            {
                $_SESSION['blad']='<span style="font-size:20px; color:red">Nieprawidłowy login lub
hasło</span>';
                $_SESSION['haslo2']=$haslo_md5;
                header('Location: index.php');
            }

            $conn->close();
        }
    }
?>

```

Rysunek 42: Skrypt logujący

Kolejnym zabezpieczeniem jest także timeout i automatyczne wylogowanie po przekroczeniu dopuszczalnego czasu bezczynności. Jest to przydatne w sytuacji, gdy nauczyciel wychodząc z sali zapomni się wylogować z dziennika. Jest to także narzędzie stosowane bardzo powszechnie.



Rysunek 43: Powiadomienie o przekroczeniu timeouta - wylogowanie

### 5.4.3 Implementacja funkcjonalności

Poniżej znajdują się przykłady skryptów PHP odpowiedzialnych za daną funkcjonalność w systemie:

W większości zakładek dane pobierane z bazy są umieszczane w budowanej na bieżąco tabeli, która umożliwia wyświetlenie ich w dobrze wyglądający sposób. Jest to wykorzystywane zarówno przy ocenach, frekwencji, jak i lekcjach (temat i sprawdzanie obecności). Skrypt 45 ukazuje budowę tabeli ocen uczniów u nauczyciela. W pozostałych przypadkach zostało to zrobione analogicznie.

3b wychowanie fizyczne ▾

Nr	Imie i nazwisko	Oceny	
1.	MARIA BEDNAREK		+
2.	WERONIKA KANIA	2	+
3.	MICHALINA MAJ		+
4.	ŁUKASZ MAJEWSKA		+
5.	LEOKADIA SADOWSKI		+
6.	HUBERT SAWICKI	4	+
7.	WIKTORIA SZULC		+
8.	CEZARY WIERZBICKI		+
9.	WOJCIECH WILK		+

Rysunek 44: Przykładowa tabela z ocenami, możliwością dodania i edycji



```

<?php
    if(isset($_SESSION['klasa_do_pokazu']) and $_SESSION['klasa_do_pokazu'] != 0)
    {

        $conn=new mysqli("localhost", "id13767441_dziennik", "bu#@!PEwH'DgS7cp",
"id13767441_dziennik");

        $k_id = $_SESSION['klasa_do_pokazu'];

        $result = $conn->query("CALL lista_osob_w_klasie($k_id)");
        $conn->close();

        $conn=new mysqli("localhost", "id13767441_dziennik", "bu#@!PEwH'DgS7cp",
"id13767441_dziennik");
        $k_id = $_SESSION['klasa_do_pokazu'];

        $result4 = $conn->query("CALL oceny_nauczyciela_w_klasie($n_id,$k_id)");
        $conn->close();


        echo "<table border=1 style='font-size:15px;'>";
        if($result->num_rows > 0) {
            $nr = 1;

            echo "<tr><td>";
            echo "Nr";
            echo "</td><td>";
            echo "Imie i nazwisko";
            echo "</td><td id='ocena' colspan=3>";
            echo "Oceny";

            echo "</td></tr>";

            $ucznen = $result4->fetch_assoc();
            while($row = $result->fetch_assoc()) {
                echo "<tr style='height:30px;'><td style='font-size:15px;'>";
                echo $nr . '.';
                echo "</td><td style='font-size:15px;'>";
                echo $row['imie'];
                echo " ";
                echo $row['nazwisko'];
                echo "</td>";
                $ucz_id = $row['uczen_ID'];
                $imie = $row['imie'];
                $nazw = $row['nazwisko'];
                $i = 0;
                if(isset($ucznen['uczen_ID']) and $row['uczen_ID'] == $ucznen['uczen_ID'])
                {

                    while( isset($ucznen['uczen_ID']) and $row['uczen_ID'] ==

$ucznen['uczen_ID'])
                    {

                        echo "<td headers='ocena' style='width:20px;font-size:15px;'>";
                        echo "<button style='width:20px;font-size:15px;'
onClick='openForm2(\"".$_ucznen['ocena_ID']."\", \"".$_imie."\", \"".$_nazw.\"\", \"".$_ucznen['stopien'] . "\", \"".$_ucznen['waga'] . "\", \"".$_ucznen['data'] . "\",
\"".$_ucznen['opis'] . "\")' >

                            <div class=\"tooltip\">
                                \"".$_ucznen['stopien'].\"
                                <span class=\"tooltiptext\">
                                    \"".$_ucznen['stopien'].\"<br>".$_ucznen['waga'].\"
<br>".$_ucznen['opis'].\"<br>".$_ucznen['data'].\"
                                </span>
                            </div></button>";
                        echo "</td>";
                        $ucznen = $result4->fetch_assoc();
                        $i = $i +1;
                        ##\"".$_ucznen['ocena_ID'].\"\", \"".$_imie.\"\", \"".$_nazw.\"\",
\"".$_ucznen['stopien'] . "\", \"".$_ucznen['waga'] . "\", \"".$_ucznen['data'] . "\", \"".$_ucznen['opis'] . "\"

$ucznen['opis'] . "\"

                    }

                }

                while( $i < 30 )
                {
                    echo "<td headers='ocena' style='width:20px;font-size:15px;'></td>";
                    $i = $i + 1;
                }

                $np_id = $_SESSION['np_klasy'];
                echo "<td><button
onClick='openForm(\"".$_ucz_id.\"\", \"".$_imie.\"\", \"".$_nazw.\"\", \"".$_

style='display:inline-block;'>
+
</button></td>";

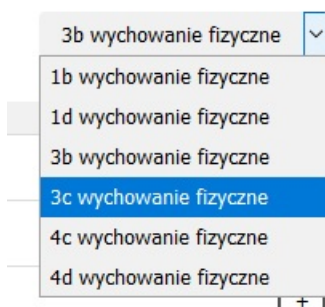
                echo "</tr>";
                $nr = $nr + 1;
            }
        }
        echo "</table>";
    }
}
?>
```

Rysunek 45: Skrypt budujący tabelę z ocenami

Kolejnym ważnym skryptem często wykorzystywanym w aplikacji jest lista typu combo box. Służy zarówno nauczycielowi do wybierania przedmiotu i klasy oraz rodzicowi do wyboru dziecka.

```
<form method="POST" >
    <select id="cmbMake" name="Make" onchange="onChangeCmb()">
        <?php
            echo '<option value = 0 > ---Wybierz klasę--- </option>';
            while($klasy = $result3 ->fetch_assoc() )
            {
                $id_klasy =$klasy['klasa_ID'];
                $np_id = $klasy['nauczyciel_przedmiotu_ID'];
                $values = '{"k_id":"' . $id_klasy . '","np_id":"' . $np_id . ' '}';
                echo '<option value=' . $values . '>' . $klasy['oddzial'] . '
                ' . $klasy['przedmiot_nazwa'] . '</option>';
            }
        <?php
        </select>
    </form>
```

Rysunek 46: Skrypt budujący listę typu combo box



Rysunek 47: Lista typu combo box

Często w aplikacji pojawia się okienko, w którym można coś wpisać. Do tego celu wykorzystano skrypt wykonywany po stronie użytkownika (javascript) wraz z biblioteką jquery. Poniższy przykład przedstawia okienko do wpisywania ocen przez nauczyciela.

```

<script src="http://code.jquery.com/jquery-1.9.1.js"></script>
<script language="javascript" type="text/javascript">

function openForm2(id, imie, nazwisko, ocena, waga, data, opis ) {
    //id, imie, nazwisko, ocena, waga, data, opis
    $oc_id = id;
    document.getElementById("form2_uczen").innerHTML = imie + ' ' + nazwisko;

    document.getElementById("form2_data").innerHTML = data;
    document.getElementById("opis_oceny2").innerHTML = opis;

    document.getElementById("select2_stopien").value = ocena;
    document.getElementById("select2_waga").value = waga;

    document.getElementById("myForm2").style.display = "block";
}

function closeForm2() {
    document.getElementById("myForm2").style.display = "none";
}

function edytuj() {

var element_stopien2 = document.getElementById("select2_stopien");
var s2 = element_stopien2.options[element_stopien2.selectedIndex].text;
window.alert($oc_id);
var element_waga2 = document.getElementById("select2_waga");
var w2 = element_waga2.options[element_waga2.selectedIndex].text;
$.ajax({
    url: "edytujOcene.php",
    data: {
        ocena_id: $oc_id,
        stopien: s2,
        waga: w2,
        opis: $('#opis_oceny2').val()
    },
});
window.alert("Ocena została edytowana");
document.getElementById("myForm2").style.display = "none";
}

function openForm(id, imie, nazwisko, np_id) {

    $ucz_id = id;
    $np_id = np_id;
    document.getElementById("form_uczen").innerHTML = imie + ' ' + nazwisko;
    document.getElementById("myForm").style.display = "block";
}

function closeForm() {
    document.getElementById("myForm").style.display = "none";
}

function dodaj() {

var element_stopien = document.getElementById("select_stopien");
var s = element_stopien.options[element_stopien.selectedIndex].text;
window.alert(s);
var element_waga = document.getElementById("select_waga");
var w = element_waga.options[element_waga.selectedIndex].text;
$.ajax({
    url: "dodajOcene.php",
    data: {
        stopien: s,
        waga: w,
        opis: $('#opis_oceny').val(),
        uczen_id: $ucz_id,
        np_id: $np_id
    },
});
window.alert("Ocena została dodana");
document.getElementById("myForm").style.display = "none";
}

function onChangeCmb() {

var element = document.getElementById("cmbMake");
var value = element.options[element.selectedIndex].value;
var v = JSON.parse(value);
$.ajax({
    url: "../wyborklasy.php",
    data: {
        id_klasy: v.k_id,
        np_klasy: v.np_id
    },
});
setTimeout(function(){ window.location.reload(true); }, 100);
}

</script>

```

Rysunek 48: Skrypt zarządzający okienkiem do wpisywania ocen

Rysunek 49: Okno do wpisywania ocen przez nauczyciela

W każdej podstronie aplikacji dostępne jest proste menu z zakładkami. Są one budowane dynamicznie tj. każda z nich ma odpowiednią wielkość w zależności od długości tekstu. Poniżej przedstawiono przykład zakładek u ucznia:

```
<div id="container">

  <div id="logo">

    <h1>Zalogowano jako uczeń</h1>

  </div>

  <a href="u_konto.php">
    <div id="inne">
      Zarządzanie kontem
    </div>
  </a>

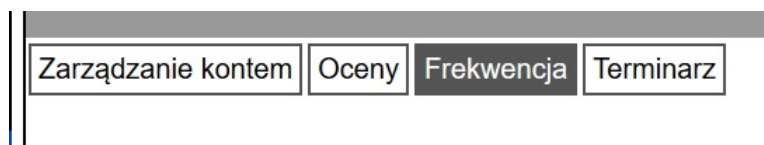
  <a href="u_oceny.php">
    <div id="inne">
      Oceny
    </div></a>

  <a href="u_frekwencja.php">
    <div id="teraz">
      Frekwencja
    </div> </a>

  <a href="u_terminarz.php">
    <div id="inne">
      Terminarz
    </div>
  </a>

</div>
```

Rysunek 50: Skrypt pokazujący zakładki menu



Rysunek 51: Zakładki menu u ucznia

## 6 Podsumowanie i wnioski

Podsumowując, po wielu problemach i próbach udało się zbudować działający system bazodanowy składający się z bazy działającej na silniku MariaDB oraz przeglądarkowej aplikacji dostępowej korzystającej z języka PHP. Projekt ten co prawda nie stworzył w pełni funkcjonalnego, dokończonego i gotowego do implementacji dziennika elektronicznego, ale solidną podstawę, która po odpowiednim nakładzie (głównie czasowym, trochę też zasobowym) mogłaby się nim stać. Dzięki temu wiemy, jak działa większość aplikacji webowych, z którymi spotykamy się na każdym kroku. Na pewno jest to była to lepsza koncepcja niż rozwiązywanie list zadań z SQL, czego na samym początku się obawialiśmy.

## Spis rysunków

1	Model konceptualny encji dziennika elektronicznego . . . . .	7
2	Model logiczny relacji dziennika elektronicznego . . . . .	8
3	Model znormalizowany relacji dziennika elektronicznego . . . . .	9
4	Kod SQL widoku . . . . .	10
5	Wywołanie widoku i wynik operacji . . . . .	10
6	Widok tworzący listę adresów . . . . .	10
7	Widok tworzący listę lekcji . . . . .	11
8	Widok tworzący listę nauczycieli . . . . .	11
9	Widok tworzący listę ocen . . . . .	11
10	Widok tworzący listę rodziców . . . . .	11
11	Kod SQL tworzący procedurę . . . . .	12
12	Wywołanie procedury i wynik operacji . . . . .	12
13	Diagram przypadków użycia związanych z usprawiedliwianiem nieobecności (edycją frekwencji) . . . . .	13
14	Diagram przypadków użycia związanych z ocenianiem i terminarzem . . . . .	14
15	Diagram przypadków użycia związanych z wystawianiem i odczytem uwag . . . . .	14
16	Diagram przypadków użycia związanych z administratorem systemu bazodanowego . . . . .	15
17	Ekran logowania . . . . .	16
18	Zakładki widoczne z poziomu konta administratora . . . . .	16
19	Zakładki widoczne z poziomu konta nauczyciela . . . . .	17
20	Zakładki widoczne z poziomu konta rodzica . . . . .	17
21	Zakładki widoczne z poziomu konta ucznia . . . . .	18
22	Lista zaimplementowanych procedur i funkcji . . . . .	19
23	Lista zaimplementowanych tabel i widoków . . . . .	20
24	Okno programu XAMPP po uruchomieniu Apache'a i MySQL'a . . . . .	21
25	Folder htdocs po wklejeniu plików . . . . .	21
26	Tworzenie bazy danych . . . . .	22
27	Importowanie systemu bazodanowego . . . . .	22
28	Dodawanie konta użytkownika . . . . .	23
29	Ekran logowania . . . . .	23
30	Po zalogowaniu . . . . .	24
31	Podgląd ocen . . . . .	24
32	Podgląd frekwencji . . . . .	25

33	Zakładka ocen z perspektywy nauczyciela . . . . .	25
34	Lista wyboru jednej z uczonych klas . . . . .	26
35	Oceny klasy i otwarte okno dodawania oceny . . . . .	26
36	Oceny klasy po dodaniu oceny . . . . .	26
37	Domyślna zakładka ucznia . . . . .	27
38	Zakładka oceny z perspektywy ucznia . . . . .	27
39	Szczegóły oceny . . . . .	28
40	Fragment widoku <code>ocena_full_info</code> ukazujący wpisaną ocenę . . . . .	28
41	Skrypt łączący się z bazą danych - przykład . . . . .	29
42	Skrypt logujący . . . . .	30
43	Powiadomienie o przekroczeniu timeouta - wylogowanie . . . . .	31
44	Przykładowa tabela z ocenami, możliwością dodania i edycji . . . . .	31
45	Skrypt budujący tabelę z ocenami . . . . .	32
46	Skrypt budujący listę typu combo box . . . . .	33
47	Lista typu combo box . . . . .	33
48	Skrypt zarządzający okienkiem do wpisywania ocen . . . . .	34
49	Okno do wpisywania ocen przez nauczyciela . . . . .	35
50	Skrypt pokazujący zakładki menu . . . . .	35
51	Zakładki menu u ucznia . . . . .	36