



**WYŻSZA SZKOŁA BANKOWA
w Poznaniu**

WYDZIAŁ FINANSÓW I BANKOWOŚCI

Projekt i implementacja elektronicznego dziennika szkolnego

PROJEKT DYPLOMOWY

Poznań 2022



DANE PARTNERÓW

A1. Dane Opiekuna

Imię i nazwisko	Adam Wojciechowski
Stopień / Tytuł naukowy	Doktor
Data i podpis	

A2. Dane członków Zespołu projektu

Imię i nazwisko	Jakub Fronczyk
Kierunek studiów	Informatyka
Tryb studiów	Niestacjonarne
Data i podpis	

Imię i nazwisko	Miłosz Fizyczak
Kierunek studiów	Informatyka
Tryb studiów	Niestacjonarne
Data i Podpis	

Imię i nazwisko	Adrian Trojanowski
Kierunek studiów	Informatyka
Tryb studiów	Niestacjonarne
Data i podpis	



ZAŁOŻENIA PROJEKTU

B1. Opis projektu

1. Problem badawczy

Problemem badawczym w naszym projekcie inżynierskim było zaprojektowanie i zbudowanie działającego i funkcjonalnego systemu, który mógłby być wykorzystywany do obsługi dziennika elektronicznego przez nauczycieli i uczniów. Użytkownik systemu będzie miał możliwość między innymi sprawdzania swoich ocen, a nauczyciel będzie mógł między innymi wystawiać oceny uczniom. Będzie także administrator, który będzie posiadać uprawnienia pozwalające na dodawanie nowych uczniów oraz nauczycieli.

2. Uzasadnienie wyboru tematu

W dzisiejszych czasach coraz popularniejsze stają się dzienniki elektroniczne. W dobie cyfryzacji szkoły odchodzą od dawnych metod dokumentowania ocen i uwag uczniów. Rozwiązania oparte o platformy do zarządzania wszystkimi danymi, które niegdyś były dostępne tylko w fizycznym dzienniku, który łatwo można było uszkodzić lub zgubić stają się rzeczywistością, a dane dostępne na serwerze pozwalają w dowolnej chwili sprawdzić status ocen lub zaktualizować je. Takie rozwiązanie jest o wiele trwalsze i bezpieczniejsze, a co więcej pozwala nam na dostęp do naszych profili z dowolnego miejsca w dowolnej chwili.

3. Zakres podmiotowy, przedmiotowy, czasowy i przestrzenny

Elektroniczny dziennik szkolny jest skierowany do szkół, które jeszcze nie zrezygnowały z tradycyjnego sposobu dokumentowania. Zakładamy, że ułatwi zarządzanie uczniami i ich ocenami, a uczniom umożliwi stały dostęp do ich danych dzięki przejrzystemu dla użytkownika interfejsowi graficznemu oraz sprawnemu systemowi zarządzania danymi.

4. Metodyka pracy (metody i techniki badawcze)

W naszym projekcie dyplomowym korzystaliśmy z metod jakościowych. Użyliśmy ich do budowy naszego elektronicznego dziennika, w którym użytkownicy mają możliwość zarządzania swoimi danymi, gdzie najważniejszą cechą takiego systemu powinna być niezawodność. Z tego też względu w naszej pracy w głównej mierze postanowiliśmy postawić na jakość i niezawodność oprogramowania. W szkołach, w których



zapropowaliśmy wprowadzenie dziennika elektronicznego po miesiącu przeprowadziliśmy ankietę dla użytkowników oprogramowania aby dowiedzieć się jaka jest ich opinia. Nasze badanie przeprowadziliśmy na grupie 7 nauczycieli i 23 uczniów. Udział w ankiecie wzięły osoby, które były obecne w szkole w dniu przeprowadzenia badania. Po przeprowadzeniu analizy badania mogliśmy dowiedzieć jak dziennik elektroniczny został przyjęty przez osoby, które wcześniej nie miały możliwości funkcjonować w takim systemie. Pytania, które zawarliśmy w kwestionariuszu zawierały odpowiedzi jednokrotnego wyboru tak aby uzyskać konkretną informację na dany temat i pozyskać jak najwięcej nowych danych aby poszerzać informacje o funkcjonowaniu dzienników elektronicznych w szkołach.

B2. Cele projektu

Cel nadrzędny

Celem nadrzędnym projektu było utworzenie dziennika elektronicznego, dzięki któremu użytkownik w łatwy i przejrzysty sposób można zarządzać danymi. Nauczyciel będzie mógł z łatwością dodawać oceny do konkretnych uczniów, a uczeń będzie miał możliwość bezproblemowego dostępu do swoich danych (m.in. ocen). Pod uwagę wzięliśmy fakt, że nauczyciele i uczniowie bardzo często są zmuszeni pracować w warunkach domowych. Stworzony przez nas system ma ułatwić im funkcjonowanie i usprawniać przekaz informacji na odległość.

Cele szczegółowe

- Stworzenie przejrzystego, łatwego w użyciu systemu zarządzania ocenami dla użytkowników;
- Stworzenie responsywnej strony, która będzie bezproblemowo łączyć się z bazą danych i będzie skierowana głównie do użytkowników korzystających z komputera.
- Zaprojektowanie i utworzenie systemu logowania oraz rejestracji użytkowników;
- Utworzenie systemu dodawania uczniów oraz nauczycieli do bazy danych strony internetowej poprzez panel administratora;
- Zrobienie systemu wystawiania ocen uczniom przez zalogowanych użytkowników z uprawnieniami nauczyciela lub administratora.



- Umożliwienie uczniom dostępu do ich danych w tym ocen wystawianych przez nauczycieli.

B3. Ryzyka związane z realizacją projektu

Zwróciliśmy dużą uwagę na ryzyko, jakie możemy napotkać podczas realizacji pracy inżynierskiej. Jest to bardzo istotny element, gdyż analiza możliwego ryzyka pozwoliłaby zapobiec uniemożliwieniu wykonania projektu lub utrudnić jego realizację.

Największą obawą w realizacji projektu było niedotrzymanie terminów na zrealizowanie zaplanowanych zadań. W związku z wykonywaniem projektu w grupie trzyosobowej problemem był natłok pracy, a tematyka projektu opierała się na zagadnieniach tylko częściowo nam znanych. Wiązało się to z tym, że dużą część czasu poświęcaliśmy na naukę i szlifowanie wiedzy na temat nowych technologii, które były wymagane w naszym projekcie, zwłaszcza języka programowania PHP, niż na faktyczną realizację pracy inżynierskiej.

REALIZACJA

C1. Zadania w projekcie

Zadanie 1

Nazwa zadania	Utworzenie ankiety dla potencjalnych zainteresowanych elektronicznym dziennikiem szkolnym
Podmioty zaangażowane w realizację zadania	Miłosz Fizyczak
Krótki opis zadania (<i>maks. 5 zdań</i>)	Zadanie miało na celu utworzenie ankiety, która pomoże nam poznać opinię potencjalnych użytkowników i dzięki której będziemy mogli lepiej zaplanować nasze dalsze zadania.
Efekty realizacji zadania (opracowania/ decyzje / dokumentacja techniczna itp.)	Efektem realizacji zadania jest dołączona ankieta
Termin rozpoczęcia realizacji zadania	10.05.2021
Termin zakończenia realizacji zadania	12.06.2021



Zadanie 2

Nazwa zadania	Zaprojektowanie wyglądu strony internetowej
Podmioty zaangażowane w realizację zadania	Adrian Trojanowski
Krótki opis zadania (<i>maks. 5 zdań</i>)	Zadanie miało na celu zaprojektowanie ogólnego wyglądu strony w tymi rozmieszczenia poszczególnych przycisków oraz elementów wyboru kategorii, a także okien logowanie oraz dodawania danych użytkowników.
Efekty realizacji zadania (opracowania/ decyzje / dokumentacja techniczna itp.)	
Termin rozpoczęcia realizacji zadania	04.09.2021
Termin zakończenia realizacji zadania	02.10.2021

Zadanie 3

Nazwa zadania	Zaprojektowanie funkcjonalności aplikacji z wykorzystaniem diagramu UML
Podmioty zaangażowane w realizację zadania	Milosz Fizyczak
Krótki opis zadania (<i>maks. 5 zdań</i>)	Zaplanowanie i zaprojektowanie diagramu przypadków użycia.
Efekty realizacji zadania (opracowania/ decyzje / dokumentacja techniczna itp.)	
Termin rozpoczęcia realizacji zadania	03.10.2021
Termin zakończenia realizacji zadania	06.10.2021



Zadanie 4

Nazwa zadania	Zakodowanie wyglądu strony z wykorzystaniem języka HTML, CSS, JavaScript oraz Frameworku Bootstrap
Podmioty zaangażowane w realizację zadania	Kuba Fronczyk, Adrian Trojanowski
Krótki opis zadania (<i>maks. 5 zdań</i>)	Zadanie miało na celu zaimplementowanie wyglądu strony internetowej w języku znaczników HTML z wykorzystaniem kaskadowych arkuszy stylów (CSS)
Efekty realizacji zadania (opracowania/ decyzje / dokumentacja techniczna itp.)	
Termin rozpoczęcia realizacji zadania	<i>11.10.2021</i>
Termin zakończenia realizacji zadania	21.10.2021

Zadanie 5

Nazwa zadania	Implementacja systemu zarządzania użytkownikami (logowanie oraz rejestracja)
Podmioty zaangażowane w realizację zadania	Kuba Fronczyk
Krótki opis zadania (<i>maks. 5 zdań</i>)	Zadanie miało na celu utworzenie skryptu w języku PHP, który pozwoli na rejestrację oraz logowanie użytkowników. Całe zadanie miało zostać zrealizowane przy wykorzystaniu bazy danych MySQL.
Efekty realizacji zadania (opracowania/ decyzje / dokumentacja techniczna itp.)	
Termin rozpoczęcia realizacji zadania	<i>25.10.2021</i>
Termin zakończenia realizacji zadania	27.11.2021



Zadanie 6

Nazwa zadania	Zaprojektowanie i zaimplementowanie systemu wyboru kategorii u użytkowników
Podmioty zaangażowane w realizację zadania	Kuba Fronczyk
Krótki opis zadania (<i>maks. 5 zdań</i>)	Zadanie miało na celu utworzenie systemu, w którym administrator będzie mieć możliwość dodawania użytkowników do systemu, a poszczególni użytkownicy będą mieli dostęp do swoich danych w tym między innymi ocen.
Efekty realizacji zadania(opracowania/ decyzje / dokumentacja techniczna itp.)	
Termin rozpoczęcia realizacji zadania	<i>27.11.2021</i>
Termin zakończenia realizacji zadania	21.12.2021

Zadanie 7

Nazwa zadania	Zaprojektowanie bazy danych SQL oraz wykonanie stosownych diagramów.
Podmioty zaangażowane w realizację zadania	Kuba Fronczyk, Miłosz Fizyczak
Krótki opis zadania (<i>maks. 5 zdań</i>)	Zadanie polegało na zaprojektowaniu bazy danych, która będzie odpowiednia do wymagań systemu. Do bazy danych należało utworzyć stosowne diagramy.
Efekty realizacji zadania(opracowania/ decyzje / dokumentacja techniczna itp.)	
Termin rozpoczęcia realizacji zadania	<i>13.10.2021</i>
Termin zakończenia realizacji zadania	15.11.2021



Zadanie 8

Nazwa zadania	Utworzenie dokumentacji do projektu opisującej działanie całego systemu.
Podmioty zaangażowane w realizację zadania	Miłosz Fizyczak, Kuba Fronczyk, Adrian Trojanowski
Krótki opis zadania (<i>maks. 5 zdań</i>)	Zadanie miało na celu opisanie poszczególnych etapów tworzenia projektu, dodanie diagramów oraz zrzutów ekranu obrazujących działanie całego systemu.
Efekty realizacji zadania (opracowania/ decyzje / dokumentacja techniczna itp.)	
Termin rozpoczęcia realizacji zadania	<i>10.01.2022</i>
Termin zakończenia realizacji zadania	<i>11.02.2022</i>



C2.Realizacja projektu

1. Założenia teoretyczne

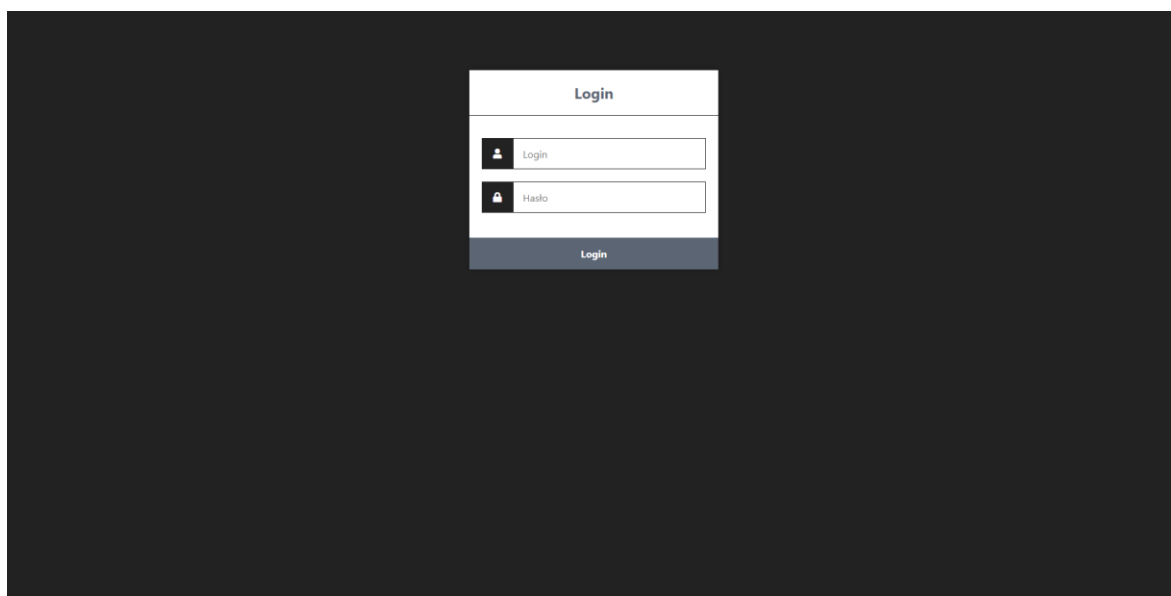
Wprowadzenie elektronicznych dzienników stało się nieodłącznym elementem systemu szkolnictwa. Celem naszego badania było poznanie opinii osób, które bezpośrednio korzystały z naszej wersji dziennika elektronicznego, oraz sprawdzenie jak wpływa na pracę nauczycieli i komunikację z uczniami. Jak wiadomo obecnie na rynku zostało już wprowadzone kilka systemów o takich funkcjach. Naszym celem było wprowadzenie nowego systemu, który wyróżniałby się na tle już istniejących systemów szczególną łatwością obsługi, przejrzystym interfejsem oraz tylko niezbędnymi funkcjami przeznaczonymi dla użytku uczniów i nauczycieli. Z ankiety wynikało, że większość badanych chce korzystać z dziennika w formie elektronicznej .

Badania empiryczne/inne

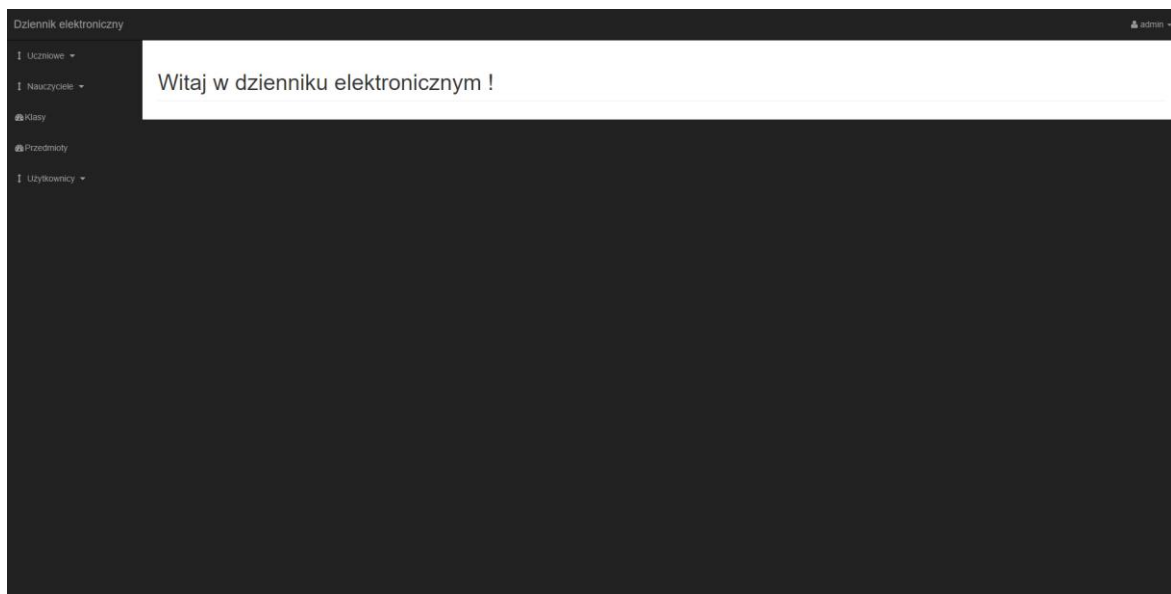
Zdecydowaliśmy się na przeprowadzenie badań w szkole, w której zostało wprowadzone nasze oprogramowanie w ramach miesięcznego testowania dziennika elektronicznego przez nauczycieli i uczniów, którzy wcześniej nie mieli okazji korzystać z takiego systemu. Wyniki ankiety pozwoliły nam dowiedzieć się jak dziennik jest postrzegany ze strony nauczycieli i uczniów dostarczając nam cennych informacji wykazujących zainteresowanie taką formą prowadzenia dziennika . Zauważyliśmy znaczne zainteresowanie prowadzenia dziennika w formie elektronicznej co może w przyszłości wpłynąć na całkowitą rezygnację z dziennika papierowego według ankietowanych.

2. Wygląd dziennika elektronicznego

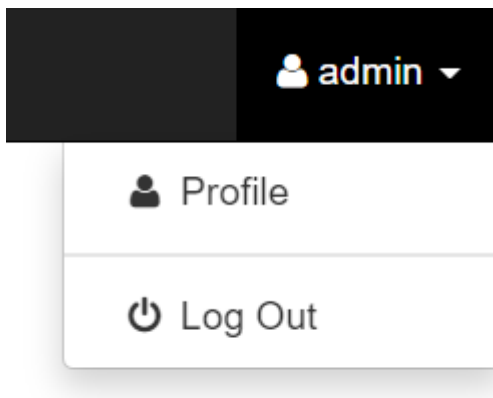
Po otwarciu dziennika w pierwszej kolejności zauważymy okno logowania, które poprosi nas o wpisanie danych takich jak login i hasło. W tym miejscu możemy się zalogować jako uczeń, nauczyciel lub administrator. Użytkownik nie ma możliwości samodzielnej rejestracji. Za tę funkcjonalność odpowiada konto administratora, który ma odpowiednie uprawnienia do wprowadzania nowych użytkowników systemu. Strona logowania prezentuje się następująco:



Gdy już system zweryfikuje naszą tożsamość poprzez podanie prawidłowego loginu i hasła oraz uda nam się zalogować, pojawi nam się okno z powitaniem:



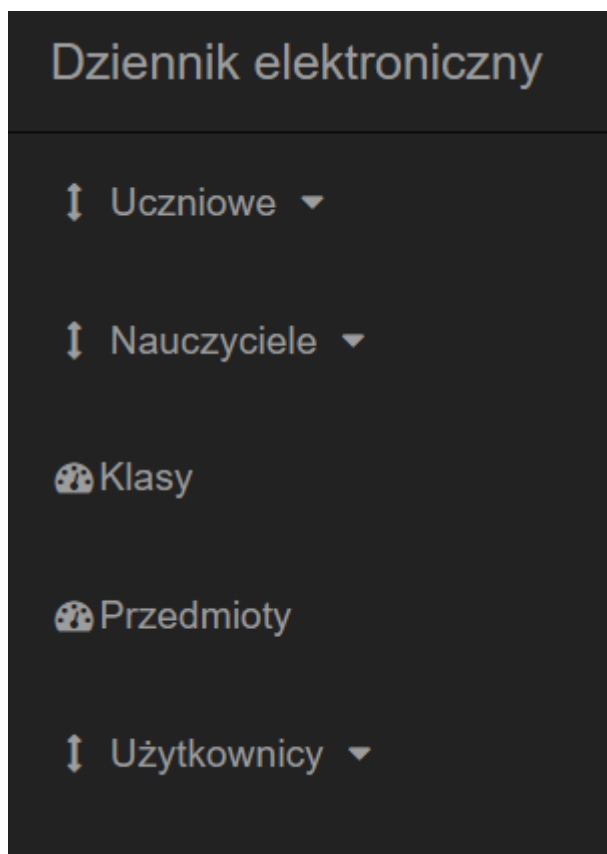
W prawym górnym rogu widzimy użytkownika, który jest aktualnie zalogowany (w tym przypadku jest to administrator). Jest to interaktywny przycisk. Gdy na niego najedziemy, będziemy mogli sprawdzić informację o profilu lub wylogować się:



Po wylogowaniu się, system przenosi nas z powrotem do okna logowania.

Z lewej strony dziennika widzimy okna wyboru z podstawowymi funkcjami takimi jak:

- Uczniowie (Administratorzy mają uprawnienia, aby dodać lub usunąć uczniów, a także wyświetlić listę wszystkich uczniów)
- Nauczyciele (Podobnie jak w przypadku uczniów, tutaj również administratorzy zarządzają użytkownikami z uprawnieniami nauczycieli, czyli mogą dodać lub usunąć nauczyciela, a także wyświetlić listę wszystkich nauczycieli)
- Klasy (W tym miejscu możemy wyświetlić listę wszystkich klas w szkole)
- Przedmioty (W tej zakładce możemy wyświetlić listę przedmiotów, które są dostępne wraz z widocznym Id przedmiotu)
- Użytkownicy (Tutaj administratorzy mogą zarządzać wszystkimi użytkownikami, także administratorami i w razie potrzeby dodać bądź usunąć osobę)



Administrator ma uprawnienia do wyświetlania wszystkich klas i może wejść w daną klasę, aby sprawdzić, jacy uczniowie są do niej przypisani oraz nimi zarządzać.





Administrator może także wyświetlać listę przedmiotów:

Dziennik elektroniczny admin

Uczniowie
Nauczyciele
Klasy
Przedmioty
Użytkownicy

Lista przedmiotów

Id przedmiotu	Przedmiot
12	Matematyka
13	Wychowanie fizyczne
14	Język polski
15	Fizyka
16	Język angielski
17	Biologia
18	Chemia
19	Wiedza o społeczeństwie
20	Historia
21	Język niemiecki

Z poziomu konta nauczyciela można wejść w kartę przedmiotu,

Dziennik elektroniczny nauczyciel

Karta przedmiotu
Uczniowie

Id klasy	Klasa	
36	1a	Wyświetl klasę
37	1b	Wyświetl klasę
38	1c	Wyświetl klasę
39	2a	Wyświetl klasę
40	2b	Wyświetl klasę
41	2c	Wyświetl klasę
42	3a	Wyświetl klasę
43	3b	Wyświetl klasę
44	3c	Wyświetl klasę

a następnie po kliknięciu w konkretną klasę, która nas interesuje wyświetlą się uczniowie przypisani do tej klasy wraz z ocenami, które są do tych uczniów przypisane. Nauczyciel ma możliwość edycji oceny u konkretnego ucznia.



Dziennik elektroniczny nauczyciel

Karta przedmiotu

Uczniowie

Id	Imię	Nazwisko	Sprawdzian	
5	Ksawery	Wójcik	5	Dodaj ocenę
14	Zuzanna	Kaźmierczak	5	Dodaj ocenę

Uczeń natomiast może wyświetlić kartę przedmiotu, który go interesuje oraz może sprawdzić swoje oceny przypisane przez nauczyciela.

Dziennik elektroniczny Ksawery Wójcik

Karta przedmiotu

Witaj w dzienniku elektronicznym, studencie!

3. Opis sytuacji faktycznej

Funkcjonalność projektu w oparciu o diagramy UML

Jedną z form przedstawienia systemu lub danego oprogramowania jest przedstawienie go w formie języka UML - graficznego przedstawienia specyfikacji systemu informatycznego. Na ich podstawie widzimy w jaki sposób działa dany system oraz jakie ma funkcjonalności.

Diagram przypadków użycia prezentuje graficznie funkcję które są widoczne przez użytkownika systemu. Na poniższym diagramie widzimy użytkowników systemu zwanych aktorami oraz interakcje które zachodzą między nim a przypadkiem użycia. Uczeń może wyświetlać oceny oraz plan lekcji. Administrator może zarządzać serwisem tworzyć nowe konta dla użytkowników. Nauczyciel ma możliwość dodawania ocen uczniom, wyświetlać listę uczniów.

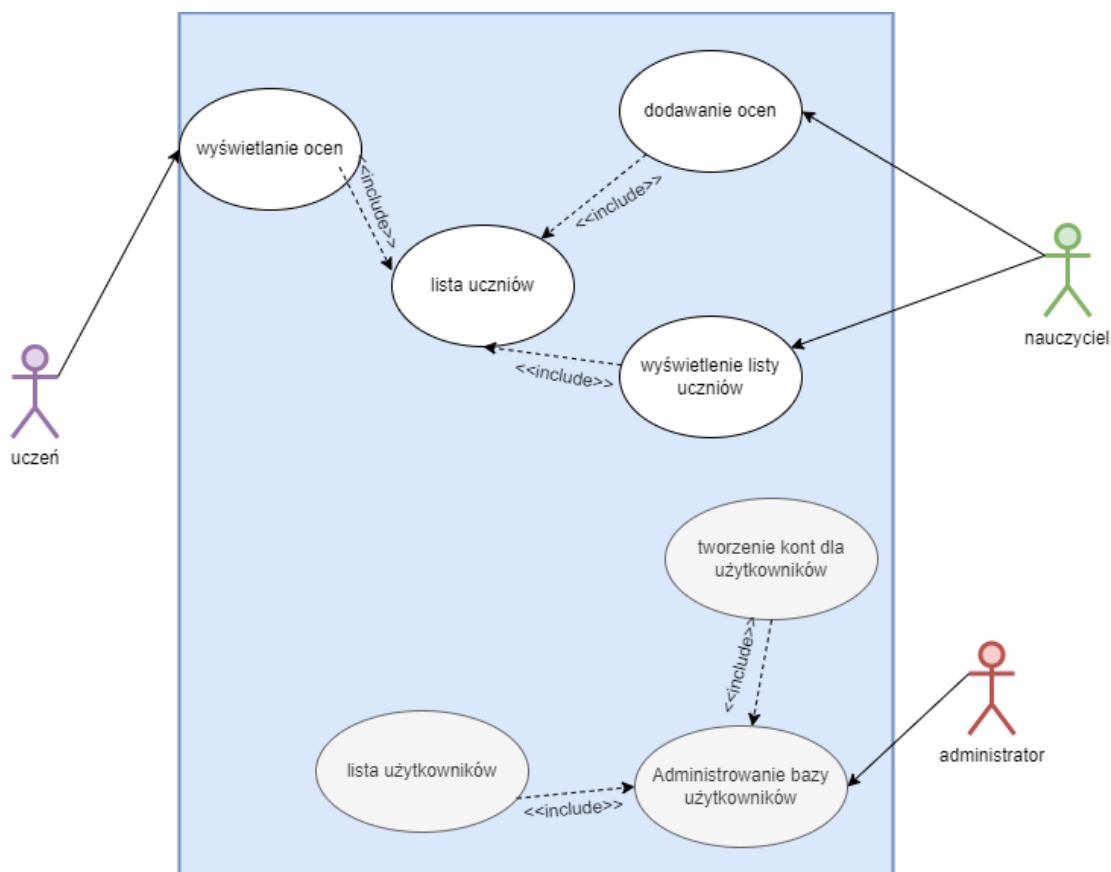
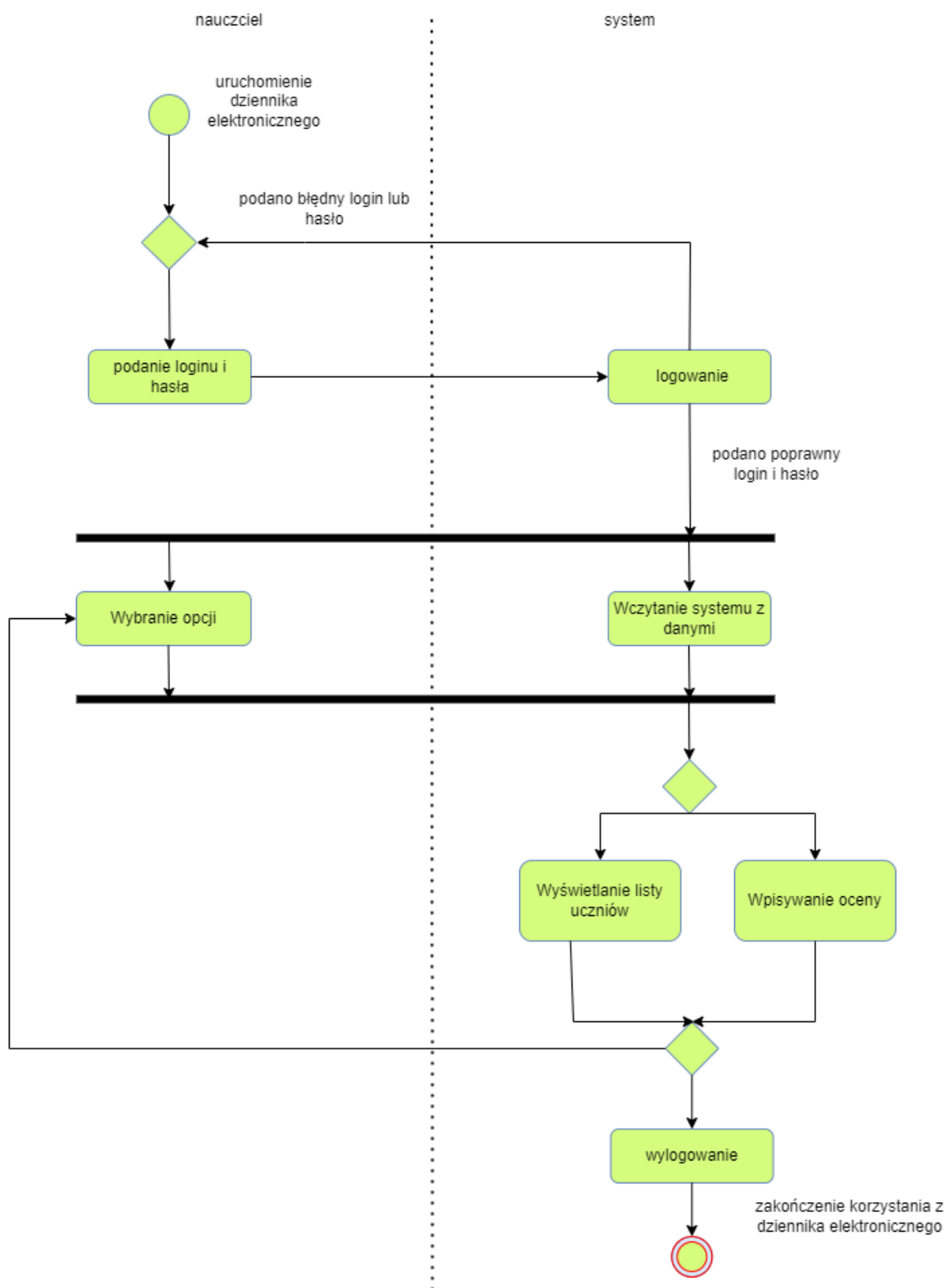


Diagram aktywności pokazuje ułamek tego co jest zawarte w systemie. Jakie kroki są wykonane w danej sytuacji przez system. Aktywnością jest logowanie się do systemu ucznia oraz możliwość wyświetlenia ocen oraz planu lekcji. System po podaniu prawidłowego loginu oraz hasła zaloguje użytkownika do systemu jeśli dane będą błędne zwróci informacje i nie pozwoli się zalogować.

System wczyta dane po poprawnym zalogowaniu oraz ukaże użytkownikowi opcje które może wybrać w systemie takie jak wyświetlenie ocen oraz opcje wylogowania z systemu tym samym zakończenie programu

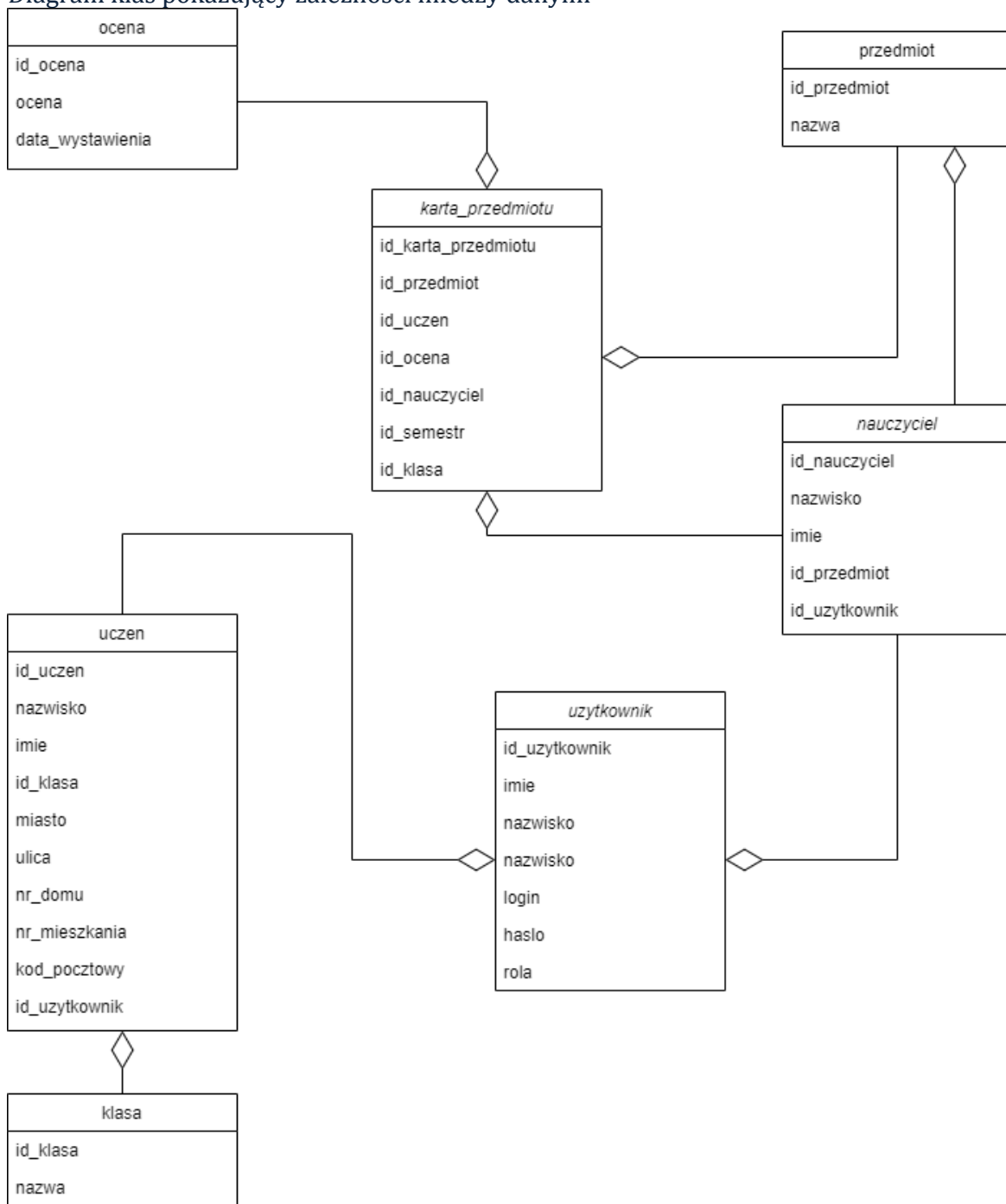


Drugim diagramem aktywności jest diagram, który obrazuje podobną sytuację jak na powyższej grafice ale z poziomu nauczyciela, gdzie nauczyciel może dodatkowo Wpisać ocenę uczniowi oraz wyświetlić listę uczniów



4. Stworzenie bazy danych, diagramu klas oraz jej implementacja

Diagram klas pokazujący zależności między danymi



Bazę danych stworzyliśmy w oparciu o język programowania PHP i baz danych MySQL. Pierwszym zadaniem było stworzenie bazy danych za pomocą narzędzia phpMyAdmin gdzie możemy gromadzić dane użytkowników systemu. W bazie zostały utworzone następujące tabele:

Karta przedmiotu

#	Nazwa	Typ
1	id_karta_przedmiotu	int(255)
2	id_przedmiot	int(255)
3	id_uczen	int(255)
4	id_ocena	int(255)
5	id_nauczyciel	int(255)
6	id_semestr	int(255)
7	id_klasa	int(11)

```
CREATE TABLE `karta_przedmiotu` (
  `id_karta_przedmiotu` int(255) NOT NULL,
  `id_przedmiot` int(255) NOT NULL,
  `id_uczen` int(255) NOT NULL,
  `id_ocena` int(255) DEFAULT NULL,
  `id_nauczyciel` int(255) NOT NULL,
  `id_semestr` int(255) NOT NULL,
  `id_klasa` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Klasa

#	Nazwa	Typ
1	id_klasa	int(255)
2	nazwa	varchar(3)

```
CREATE TABLE `klasa` (
  `id_klasa` int(255) NOT NULL,
  `nazwa` varchar(3) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Nauczyciel

#	Nazwa	Typ
1	id_nauczyciel	int(255)
2	nazwisko	varchar(30)
3	imie	varchar(30)
4	id_przedmiot	int(255)
5	id_uzytkownik	int(3)



```
CREATE TABLE `nauczyciel` (  
  `id_nauczyciel` int(255) NOT NULL,  
  `nazwisko` varchar(30) NOT NULL,  
  `imie` varchar(30) NOT NULL,  
  `id_przedmiot` int(255) NOT NULL,  
  `id_uzytkownik` int(3) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Ocena

#	Nazwa	Typ
1	id_ocena	int(255)
2	ocena	int(255)
3	data_wystawienia	date

```
CREATE TABLE `ocena` (  
  `id_ocena` int(255) NOT NULL,  
  `ocena` int(255) DEFAULT NULL,  
  `data_wystawienia` date DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Przedmiot

#	Nazwa	Typ
1	id_przedmiot	int(255)
2	nazwa	varchar(50)

```
CREATE TABLE `przedmiot` (  
  `id_przedmiot` int(255) NOT NULL,  
  `nazwa` varchar(50) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Uczeń

#	Nazwa	Typ
1	id_uczen	int(255)
2	nazwisko	varchar(255)
3	imie	varchar(255)
4	id_klasa	int(255)
5	miasto	varchar(255)
6	ulica	varchar(255)
7	nr_domu	varchar(32)
8	nr_mieszkania	int(32)
9	kod_pocztowy	varchar(255)
10	id_uzytkownik	int(3)

```
CREATE TABLE `uczen` (
  `id_uczen` int(255) NOT NULL,
  `nazwisko` varchar(255) NOT NULL,
  `imie` varchar(255) NOT NULL,
  `id_klasa` int(255) NOT NULL,
  `miasto` varchar(255) NOT NULL,
  `ulica` varchar(255) NOT NULL,
  `nr_domu` varchar(32) NOT NULL,
  `nr_mieszkania` int(32) NOT NULL,
  `kod_pocztowy` varchar(255) NOT NULL,
  `id_uzytkownik` int(3) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Użytkownik

#	Nazwa	Typ
1	id_uzytkownik 	int(3)
2	imie	varchar(30)
3	nazwisko	varchar(30)
4	login	varchar(255)
5	haslo	varchar(255)
6	rola	varchar(255)

```
CREATE TABLE `uzytkownik` (
  `id_uzytkownik` int(3) NOT NULL,
  `imie` varchar(30) NOT NULL,
  `nazwisko` varchar(30) NOT NULL,
  `login` varchar(255) NOT NULL,
  `haslo` varchar(255) NOT NULL,
  `rola` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Połączenie bazy danych jest połączone ze stroną internetową za pomocą skryptu zawartego w pliku o nazwie „db_con.php” który znajduje się w folderze „includes”.

```

1  <?php
2
3  $db['db_host'] = "localhost";
4  $db['db_user'] = "root";
5  $db['db_pass'] = "";
6  $db['db_name'] = "dziennik_elektroniczny";
7
8  foreach($db as $key => $value){
9
10     define(strtoupper($key), $value);
11
12 }
13
14
15
16
17 $connection = mysqli_connect(DB_HOST, DB_USER, DB_PASS, DB_NAME);
18
19
20
21
22 ?>

```

5. Zaimplementowanie systemu logowania.

Aby zaimplementować system logowania i użytkowników został użyty język programowania, jeden z najbardziej popularnych w web developmencie odpowiadający za tzw. backend, czyli PHP oraz MySQL pozwalający na zarządzanie relacyjnymi bazami danych.

Począwszy od utworzenia bazy danych w phpMyadmin, narzędziu wbudowanym w oprogramowanie XAMPP mogliśmy przechowywać dane naszych użytkowników takie jak: imię, nazwisko, login oraz hasło oraz rolę.

id_uzytkownik	imie	nazwisko	login	haslo	rola
18	admin	admin	admin	opMNSBg7BX	admin
19	Adam	Kowalski	adamkowalski@gmail.com	f6pUAtQsMh	uczeń
21	Patryk	Kurpiel	patrykkurpiel@gmail.com	GTxU7PVMnX	nauczyciel

Najważniejszą funkcjonalnością systemu logowania jest rozdzielenie użytkowników na role, ponieważ każda rola nadaje odpowiednie uprawnienia do tego co może robić i widzieć odpowiedni użytkownik w systemie. Na wyżej zamieszczonym zdjęciu widać, że do wyboru są trzy możliwe role admin, uczeń oraz nauczyciel. Administrator ma możliwość zarządzania prawie całym systemem. Cztery podstawowe funkcję **CRUD** (od ang. create, read, update, delete, tłum. utwórz, odczytaj, aktualizuj, usuń). Taka możliwość została stworzona odnośnie użytkowników, nauczycieli oraz uczniów jak i tworzenie użytkowników, nadawania im roli oraz przypisywania ich do poszczególnych uczniów i nauczycieli. Rola nauczyciela odpowiada za wyświetlanie uczniów oraz dodawanie ocen do poszczególnych uczniów.

```

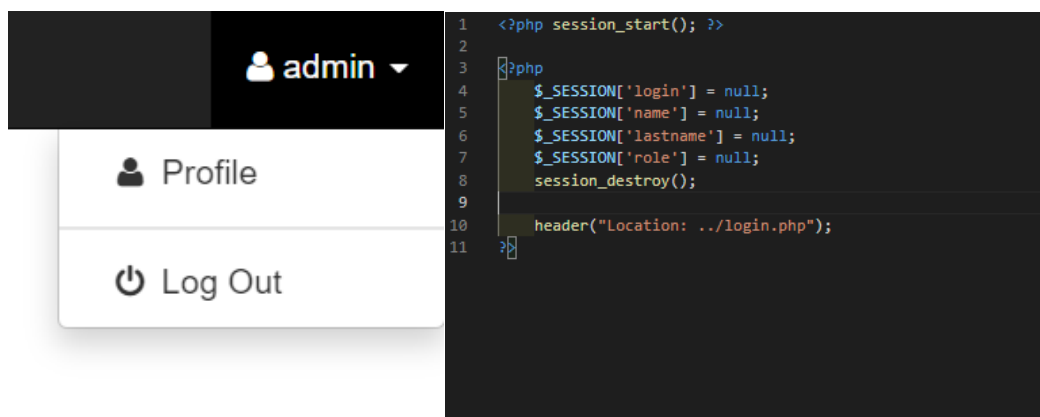
1  <?php include "db_con.php"; ?>
2  <?php session_start(); ?>
3
4  <?php
5
6  if (isset($_POST['login'])) {
7      $username = $_POST['username'];
8      $password = $_POST['password'];
9
10     $username = mysqli_real_escape_string($connection, $username);
11     $password = mysqli_real_escape_string($connection, $password);
12
13     $query = "SELECT * FROM uzytkownik WHERE login = '{$_username}' ";
14     $select_user_query = mysqli_query($connection, $query);
15
16     if (!$select_user_query) {
17         die("Error". mysqli_error($connection));
18     }
19
20     while($row = mysqli_fetch_array($select_user_query)) {
21         $db_user_id = $row['id_uzytkownik'];
22         $db_user_login = $row['login'];
23         $db_user_password = $row['haslo'];
24         $db_user_name = $row['imie'];
25         $db_user_lastname = $row['nazwisko'];
26         $db_user_role = $row['rola'];
27     }
28 }
29
30 if($username != $db_user_login && $password != $db_user_password){
31     header("Location: ../login.php");
32 }else if($username == $db_user_login && $password == $db_user_password){
33     $_SESSION['login'] = $db_user_login;
34     $_SESSION['name'] = $db_user_name;
35     $_SESSION['lastname'] = $db_user_lastname;
36     $_SESSION['role'] = $db_user_role;
37     if(isset($_SESSION['role'])){
38         if($_SESSION['role'] == 'admin'){
39             header("Location: ../admin.php");
40         }elseif($_SESSION['role'] == 'uczeń'){
41             header("Location: ../student.php");
42         }else{
43             header("Location: ../teacher.php");
44         }
45     }
46 }
47
48 }else{
49     header("Location: ../login.php");
50 }
51
52
53 }
54
55 ?>

```

Powyższy kod odpowiada za logowanie oraz sprawdzenie do jakiej roli przypisany jest dany użytkownik. Za pomocą metody POST przesyłamy dane do logowania, następnie pod zmienną \$query zapisane jest zapytanie do bazy danych jakie przesłane zostanie za pomocą funkcji mysqli_query, jeżeli z jakiegoś powodu połączenie z bazą danych nie przebiegnie pomyślnie, za pomocą die funkcja mysqli_error wyświetli nam komunikat z błędem i zakończy połączenie z bazą danych. Za pomocą funkcji if sprawdzana jest poprawność

loginu oraz hasłem. Jeżeli dane będą niepoprawne zostanie załadowana ponownie strona logowania.

Wylogowania się użytkownika odbywa się za pomocą naciśnięcia przycisku



Dane w `$_SESSION` zostają wyczyszczone, następnie za pomocą funkcji `session_destroy()` niszy wszystkie dane związane z tą sesją.

6. Wdrożenie funkcji zarządzania użytkownikami.

Administrator systemu ma możliwość dodawania, edytowania i wyświetlania listy użytkowników.

Dziennik elektroniczny admin

- Uczniowie
- Nauczyciele
- Klasy
- Przedmioty
- Użytkownicy**

Lista użytkowników

Id	Imię	Nazwisko	Login	Rola				
17	Dominik	Sikora	dsikora	nauczyciel	Zmień na ucznia	Zmień na nauczyciela	Edytuj	Usuń
18	Mariusz	Stępień	mstepien	nauczyciel	Zmień na ucznia	Zmień na nauczyciela	Edytuj	Usuń
19	Mikołaj	Zalewski	mzalewski	nauczyciel	Zmień na ucznia	Zmień na nauczyciela	Edytuj	Usuń
20	Mikołaj	Marciński	mmarciński	nauczyciel	Zmień na ucznia	Zmień na nauczyciela	Edytuj	Usuń
21	Łucjan	Czerwiński	lczerwiński	nauczyciel	Zmień na ucznia	Zmień na nauczyciela	Edytuj	Usuń
26	Antonina	Czerwińska	mczerwińska	nauczyciel	Zmień na ucznia	Zmień na nauczyciela	Edytuj	Usuń
27	Natalia	Kowalczyk	nkowalczyk	nauczyciel	Zmień na ucznia	Zmień na nauczyciela	Edytuj	Usuń
28	Natalia	Sadowska	nsadowska	nauczyciel	Zmień na ucznia	Zmień na nauczyciela	Edytuj	Usuń
29	Matylda	Baran	mbaran	nauczyciel	Zmień na ucznia	Zmień na nauczyciela	Edytuj	Usuń
30	Jolanta	Mazur	jmazur	nauczyciel	Zmień na ucznia	Zmień na nauczyciela	Edytuj	Usuń

Na powyższym zdjęciu widzimy przykładowo wprowadzoną listę użytkowników o roli nauczyciela. System daje możliwość szybkiej zmiany roli z ucznia na nauczyciela oraz odwrotnie, możliwość edycji lub usunięcia nauczyciela. Poniższy fragment kodu `add_user.php` przedstawia dodawanie nauczyciela.

```

1  <?php
2
3  if(isset($_POST['create_user'])){
4
5      $user_name = $_POST['imie'];
6      $user_lastname = $_POST['nazwisko'];
7      $user_login = $_POST['login'];
8
9      <?php
10     $query = "INSERT INTO uzytkownik( imie, nazwisko, login, haslo, rola) "
11     @var string $query
12     $query .= "VALUES ('{$user_name}', '{$user_lastname}', '{$user_login}', '{$user_password}', '{$user_role}' )";
13     $create_teacher_query = mysqli_query($connection, $query);
14     if(!$create_teacher_query){
15         die("Failed to create". mysqli_error($connection));
16     }
17 }
18
19 ?>
20 <h1>Dodaj użytkownika</h1>
21 <form action="" method="post" enctype="multipart/form-data">
22     <div class="form-group">
23         <label for="imie">Imię</label>
24         <input type="text" class="form-control" name="imie">
25     </div>
26     <div class="form-group">
27         <label for="nazwisko">Nazwisko</label>
28         <input type="text" class="form-control" name="nazwisko">
29     </div>
30     <div class="form-group">
31         <label for="login">Login</label>
32         <input type="text" class="form-control" name="login">
33     </div>
34     <div class="form-group">
35         <label for="haslo">Hasło</label>
36         <input type="password" class="form-control" name="haslo">
37     </div>
38     <div class="form-group">
39         <label>Rola</label>
40         <br>
41         <select name="rola" id="">
42             <option value="uczeń">Wybierz rolę</option>
43             <option value="admin">admin</option>
44             <option value="nauczyciel">nauczyciel</option>
45             <option value="uczeń">uczeń</option>
46         </select>
47     </div>
48     <div class="form-group">
49         <input class="btn btn-primary" type="submit" name="create_user" value="Dodaj użytkownika">
50     </div>
51 </form>

```

Za pomocą metody POST przesyłamy zapytanie do bazy danych dodając dane do poszczególnych kolumn poleceniem INSERT INTO. Formularz `<form></form>` pozwala na wprowadzenie danych użytkownikowi i przesłania ich na serwer. Za pomocą `<select></select>` wyświetlają się role, które będą przypisane do użytkownika.

Wybierając edytuj w zdjęciu listy użytkowników możemy zmienić dane użytkownika.



Dziennik elektroniczny admin

- Uczniowie
- Nauczyciele
- Klasy
- Przedmioty
- Użytkownicy

Edytuj użytkownika

Imię

Nazwisko

Login

Hasło

Rola

[Edytuj użytkownika](#)

W poniższym fragmencie kodu edit_user.php zaimplementowana jest funkcjonalność zaktualizowania danych w bazie.

```
<?php
if(isset($_GET['edit_user'])){
    $edit_user_id = $_GET['edit_user'];
    $query="SELECT * FROM uzytkownik WHERE id_uzytkownik = $edit_user_id ";
    $select_user = mysqli_query($connection, $query);
    while($row = mysqli_fetch_array($select_user)){
        $user_id = $row['id_uzytkownik'];
        $user_name = $row['imie'];
        $user_lastname = $row['nazwisko'];
        $user_login = $row['login'];
        $user_password = $row['haslo'];
        $user_role = $row['rola'];
    }
}

if(isset($_POST['edit_user'])){
    $user_name = $_POST['imie'];
    $user_lastname = $_POST['nazwisko'];
    $user_login = $_POST['login'];
    $user_password = $_POST['haslo'];
    $user_role = $_POST['rola'];
    $query = "UPDATE uzytkownik SET ";
    $query .="imie = '{$user_name}', ";
    $query .="nazwisko = '{$user_lastname}', ";
    $query .="login = '{$user_login}', ";
    $query .="haslo = '{$user_password}', ";
    $query .="rola = '{$user_role}' ";
    $query .="WHERE id_uzytkownik = '{$edit_user_id}' ";
    $edit_teacher_query = mysqli_query($connection, $query);
    if(!$edit_teacher_query){
        die("Failed to create". mysqli_error($connection));
    }
    header("Location: show_user.php");
}
?>
```

Dzięki metodzie POST przesyłamy zapytanie UPDATE do bazy, i aktualizujemy każde pole w bazie nowymi danymi. Dzięki metodzie GET, możemy wyświetlić jak przedstawiono na zdjęciu edycji użytkownika wcześniej wprowadzone dane.

```
<div class="form-group">
<label>Rola</label>
<br>
<select name="rola" id="">
  <option value="uczeń"><?php echo $user_role ?></option>
  <?php
  if($user_role == 'admin'){
    echo "<option value='nauczyciel'>nauczyciel</option>";
    echo "<option value='uczeń'>uczeń</option>";
  }else if($user_role == 'uczeń'){
    echo "<option value='admin'>admin</option>";
    echo "<option value='nauczyciel'>nauczyciel</option>";
  }else if($user_role == 'nauczyciel'){
    echo "<option value='admin'>admin</option>";
    echo "<option value='uczeń'>uczeń</option>";
  }
  ?>
</select>
</div>
```

We fragmencie kodu z pliku edit_user.php, dynamicznie wyświetlamy role z bazy wyłącznie te, które nie były wcześniej ustawione.

```
1 <?php include "includes/header.php"?>
2
3 <div id="wrapper">
4 <!--Navigation -->
5 <?php include "includes/navigation.php"?>
6
7 <div id="page-wrapper">
8
9 <div class="container-fluid">
10
11 <!-- Page Heading -->
12 <div class="row">
13 <div class="col-lg-12">
14
15 <?php
16
17 if(isset($_GET['source'])){
18     $source = $_GET['source'];
19 }else{
20     $source = '';
21 }
22
23 switch($source){
24     case 'add_user';
25         include "includes/add_user.php";
26         break;
27
28     case 'edit_user';
29         include "includes/edit_user.php";
30         break;
31
32     default:
33         include "includes/view_user.php";
34
35
36 }
37
38 ?>
39 </div>
40 </div>
41
42 </div>
43
44 <?php include "includes/footer.php" ?>
```

W powyższym kodzie show_user.php instrukcja switch przenosi nas do dodawania, edycji lub usuwa konkretną edycję. Analogiczna sytuacja występuje w plikach show_teacher.php oraz show_student.php.

W poniższym fragmencie kodu view_user.php

```

17 <?php
18 $query="SELECT * FROM uzytkownik";
19 $select_user = mysqli_query($connection, $query);
20 while($row = mysqli_fetch_array($select_user)){
21
22     $user_id = $row['id_uzytkownik'];
23     $user_name = $row['imie'];
24     $user_lastname = $row['nazwisko'];
25     $user_login = $row['login'];
26     $user_role = $row['rola'];
27     echo "<tr>";
28     echo "<td>".$user_id."</td>";
29     echo "<td>".$user_name."</td>";
30     echo "<td>".$user_lastname."</td>";
31     echo "<td>".$user_login."</td>";
32     echo "<td>".$user_role."</td>";
33     echo "<td><a href='show_user.php?change_to_student={$user_id}'>Zmień na ucznia</a></td>";
34     echo "<td><a href='show_user.php?change_to_teacher={$user_id}'>Zmień na nauczyciela</a></td>";
35     echo "<td><a href='show_user.php?source=edit_user&edit_user={$user_id}'>Edytuj</a></td>";
36     echo "<td><a href='show_user.php?delete={$user_id}'>Usuń</a></td>";
37     echo "</tr>";
38 }
39 }
40
41 </tbody>
42 </table>
43 <?php // delete teacher
44 if(isset($_GET['delete'])){
45     $delete_user_id = $_GET['delete'];
46     $query = "DELETE from uzytkownik WHERE id_uzytkownik = {$delete_user_id}";
47     $delete_user_query = mysqli_query($connection, $query);
48     header("Location: show_user.php");
49 }
50
51 <?php // change to student
52 if(isset($_GET['change_to_student'])){
53     $change_user_role = $_GET['change_to_student'];
54     $query = "UPDATE uzytkownik SET rola = 'uczen' WHERE id_uzytkownik = {$change_user_role} ";
55     $change_to_student_query = mysqli_query($connection, $query);
56     header("Location: show_user.php");
57     if (!$change_to_student_query){
58         die("Could not create post" . mysqli_error($connection));
59     }
60 }
61
62 <?php // change to teacher
63 if(isset($_GET['change_to_teacher'])){
64     $change_user_role = $_GET['change_to_teacher'];
65     $query = "UPDATE uzytkownik SET rola = 'nauczyciel' WHERE id_uzytkownik = {$change_user_role} ";
66     $change_to_student_query = mysqli_query($connection, $query);
67     header("Location: show_user.php");
68 }
69 }

```

Zmienna \$user_id odpowiada id_uzytkownika w bazie, po czym system wie, którego dokładnie użytkownika edytuje lub usuwa.

Implementacja funkcjonalności dodawania ucznia i nauczyciela.

W tym podpunkcie przedstawimy po krótku implementację funkcjonalności dodawania, odczytu, zaktualizowania oraz usuwania ucznia oraz nauczyciela, ponieważ schemat działania jest podobny.

Poniższe zdjęcia przedstawiają listę uczniów oraz nauczycieli.



Dziennik elektroniczny admin

Uczniowie
Nauczyciele
Klasy
Przedmioty
Użytkownicy

Lista nauczycieli

Id	Nazwisko	Imię	Przedmiot	Użytkownik		
73	Sikora	Dominik	Matematyka	Dominik Sikora	Edytuj	Usuń
74	Stępień	Marlusz	Wychowanie fizyczne	Marlusz Stępień	Edytuj	Usuń
75	Zalewski	Mikołaj	Język polski	Mikołaj Zalewski	Edytuj	Usuń
76	Marciniak	Mikołaj	Fizyka	Mikołaj Marciniak	Edytuj	Usuń
77	Czerwiński	Lucjan	Język angielski	Lucjan Czerwiński	Edytuj	Usuń
78	Czerwińska	Antonina	Biologia	Antonina Czerwińska	Edytuj	Usuń
79	Kowalczyk	Natalia	Chemia	Natalia Kowalczyk	Edytuj	Usuń
80	Sadowska	Natalia	Wiedza o społeczeństwie	Natalia Sadowska	Edytuj	Usuń
81	Baran	Matylda	Historia	Matylda Baran	Edytuj	Usuń
82	Mazur	Jolanta	Język niemiecki	Jolanta Mazur	Edytuj	Usuń

Dziennik elektroniczny admin

Uczniowie
Nauczyciele
Klasy
Przedmioty
Użytkownicy

Lista uczniów

Id	Nazwisko	Imię	Klasa	Miasto	Ulica	Numer domu	Numer mieszkania	Kod pocztowy	Użytkownik		
5	Wójcik	Ksawery	1a	Poznań	Spokojna	20	3	61-825	Ksawery Wójcik	Edytuj	Usuń
6	Sikora	Jacek	1b	Poznań	Wizorka Leona	3	1	61-825	Sikora Jacek	Edytuj	Usuń
7	Malinowski	Kajetan	1c	Poznań	Zaborze	15	2	61-825	Kajetan Malinowski	Edytuj	Usuń
8	Andrzejewski	Ariel	2a	Poznań	Jesionowa	19	2	61-825	Ariel Andrzejewski	Edytuj	Usuń
9	Piotrowski	Czesław	2b	Poznań	Röntgena Wilhelma Konrada	27	2	61-825	Czesław Piotrowski	Edytuj	Usuń
10	Laskowski	Gniewomir	2c	Poznań	Młyńska	21	2	61-825	Gniewomir Laskowski	Edytuj	Usuń
11	Zieliński	Konstanty	3a	Poznań	Krogulczka	23	7	61-825	Konstanty Zieliński	Edytuj	Usuń
12	Lis	Juliusz	3b	Poznań	Mala	29	2	61-825	Juliusz Lis	Edytuj	Usuń
13	Makowski	Artur	3c	Poznań	Jesionowa	19	2	61-825	Artur Makowski	Edytuj	Usuń
14	Kaźmierczak	Zuzanna	1a	Poznań	Głowackiego	17	8	61-825	Zuzanna Kaźmierczak	Edytuj	Usuń
15	Andrzejewska	Zuzanna	1b	Poznań	Kościelna	21	9	61-825	Zuzanna Andrzejewska	Edytuj	Usuń
16	Wilkowska	Andrea	1c	Poznań	Tamogórska	8	3	61-825	Andrea Wilkowska	Edytuj	Usuń
17	Florencja	Błaszczak	2a	Poznań	Przyłutna	15	7	61-825	Florencja Błaszczak	Edytuj	Usuń
18	Andrzejewska	Wiktoria	2b	Poznań	Długa	27	12	61-825	Wiktoria Andrzejewska	Edytuj	Usuń
19	Makowska	Edyta	2c	Poznań	Taczanowskiego Edmunda	3	24	61-825	Edyta Makowska	Edytuj	Usuń
20	Nowak	Wioletta	3a	Poznań	Grabowa	13	4	61-825	Wioletta Nowak	Edytuj	Usuń
21	Krawczyk	Anatolia	3b	Poznań	Energetyków	26	3	61-825	Anatolia Krawczyk	Edytuj	Usuń
22	Borkowska	Franciszka	3c	Poznań	Tabaczna	13	5	61-825	Franciszka Borkowska	Edytuj	Usuń

Lista uczniów i lista nauczycieli przedstawia kolumny odpowiadające bazie danych. Takie jak imię, nazwisko, przedmiot, klasa, itp. Z poziomu listy możemy usunąć lub edytować daną pozycję. Poniżej przedstawione fragmenty kodu działają bardzo podobnie jak opisane powyżej fragmenty kodu odpowiadające za użytkowników.

```
<?php
if(isset($_POST['create_student'])){
    $last_name = $_POST['nazwisko'];
    $name = $_POST['imie'];
    $student_class = $_POST['student_class'];
    $city = $_POST['miasto'];
    $street = $_POST['ulica'];
    $house = $_POST['nr_domu'];
    $apartment = $_POST['nr_mieszkania'];
    $post_code = $_POST['kod_pocztowy'];
    $student_user = $_POST['student_user'];
    $query = "INSERT INTO uczen(nazwisko, imie, id_klasa, miasto, ulica, nr_domu, nr_mieszkania, kod_pocztowy, id_uzytkownik) ";
    $query .= "VALUES ('{$last_name}', '{$name}', '{$student_class}', '{$city}', '{$street}', '{$house}', '{$apartment}', '{$post_code}', '{$student_user}' )";
    $create_student_query = mysqli_query($connection, $query);
    if(!$create_student_query){
        die("Failed to create". mysqli_error($connection));
    }
    header("Location: show_student.php");
}
?>
```

```
<?php
if(isset($_POST['create_teacher'])){
    $last_name = $_POST['nazwisko'];
    $name = $_POST['imie'];
    $teacher_subject_id = $_POST['teacher_subject'];
    $teacher_user = $_POST['teacher_user'];
    $query = "INSERT INTO nauczyciel( nazwisko, imie, id_przedmiot, id_uzytkownik) ";
    $query .= "VALUES ('{$last_name}', '{$name}', '{$teacher_subject_id}', '{$teacher_user}' )";
    $create_teacher_query = mysqli_query($connection, $query);
    if(!$create_teacher_query){
        die("Failed to create". mysqli_error($connection));
    }
    header("Location: show_teacher.php");
}
?>
```

Dzięki metodzie POST wysyłają zapytania do bazy danych.

7. Funkcja wyświetlania z panelu nauczyciela uczniów przypisanych do klasy oraz ocen

Poniższe zdjęcie przedstawia wyświetloną klasę i przypisanych do nich uczniów.

Dziennik elektroniczny
nauczyciel

Karta przedmiotu
Uczniowie

Id	Imię	Nazwisko	Sprawdzian 1	
5	Klaavery	Wójcik	5	Dodaj ocenę
14	Zuzanna	Kaźmierczak	5	Dodaj ocenę

Poniższy kod prezentuje funkcjonalność przedstawioną na powyższym zdjęciu.

```

1 <table class="table table-bordered table-hover">
2   <thead>
3     <tr>
4       <th>Id</th>
5       <th>Imię</th>
6       <th>Nazwisko</th>
7       <th>Sprawdzian 1</th>
8     </tr>
9   </thead>
10  <tbody>
11    <?php
12      if(isset($_GET['c_id'])){
13        $get_student_id = $_GET['c_id'];
14      }
15      $query="SELECT * FROM uczen WHERE id_klasa = $get_student_id";
16      $select_student_class = mysqli_query($connection, $query);
17
18      while($row = mysqli_fetch_array($select_student_class)){
19
20        $student_id = $row['id_uczen'];
21        $student_name = $row['imie'];
22        $student_lastname = $row['nazwisko'];
23        $student_grade = $row['sprawdzian_1'];
24        echo "<tr>";
25        echo "<td>".$student_id."</td>";
26        echo "<td>".$student_name."</td>";
27        echo "<td>".$student_lastname."</td>";
28
29        $query = "SELECT * FROM ocena WHERE id_ocena = {$student_grade}";
30        $select_grade = mysqli_query($connection, $query);
31        while ($row = mysqli_fetch_assoc($select_grade)){
32          $grade_id = $row['id_ocena'];
33          $grade_title = $row['ocena'];
34        }
35        echo "<td>".$student_grade."</td>";
36        echo "<td><a href='show_user.php?source=edit_teacher&t_id={$student_id}'>Dodaj ocenę</a></td>";
37        echo "<td><a href='show_subject_card.php?delete={$student_grade}'>Usuń</a></td>";
38      }
39      echo "</tr>";
40    }
41  </tbody>
42</table>
43
44<?php
45if(isset($_GET['delete'])){
46  $delete_student_grade = $_GET['delete'];
47  $query = "DELETE from uczen WHERE sprawdzian_1 = {$delete_student_grade}";
48  $delete_teacher_query = mysqli_query($connection, $query);
49}
50?>

```

Możliwość usunięcia oceny metodą GET po wyszukanie zmiennej \$student_grade, gdzie ocena odpowiada indeksowi ocenie w tabeli ocena przypisanej do tabeli ucznia.

8. Implementacja

Zakodowanie wyglądu strony z wykorzystaniem języka HTML, CSS, JavaScript oraz Frameworku Bootstrap

Do zakodowania wyglądu strony wykorzystaliśmy język HTML, CSS, JavaScript oraz Frameworku Bootstrap. Struktura strony stworzona jest w pliku admin.php to on jest główną bazą stworzenia całego szkieletu strony. W sekcji <body> - umieszczone zostały odnośniki do podstron zawarte na stronie głównej programu.

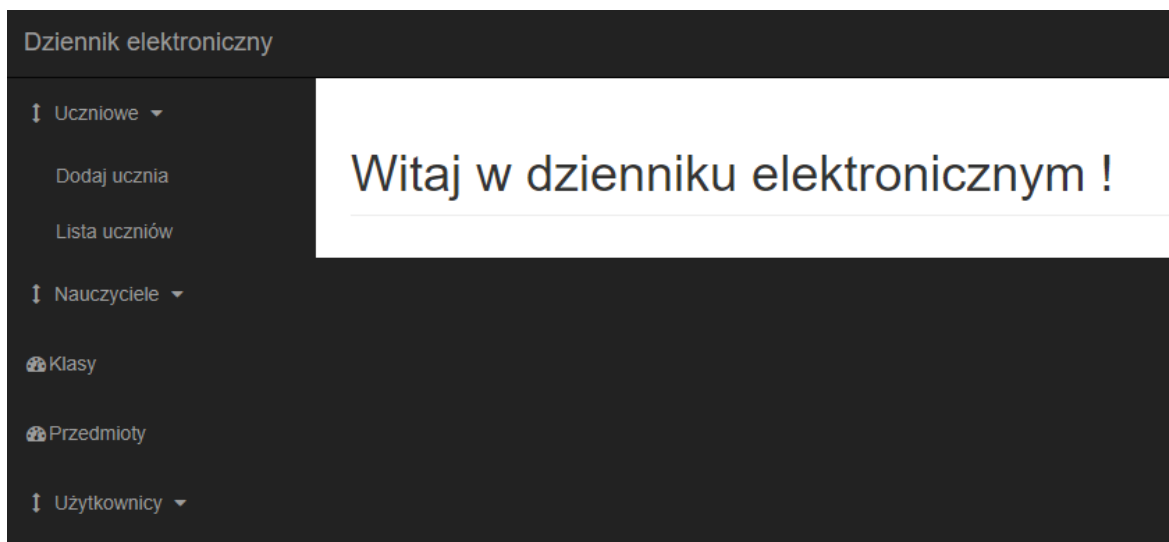
Z sposób w jaki został zaprojektowany wygląd głównej strony oraz przycisków funkcyjnych użytkowników odpowiada stosowny kod napisany z wykorzystaniem języka HTML, który można zobaczyć poniżej.

```

30 <!-- Navigation -->
31 <nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
32   <div class="navbar-header">
33     <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-ex1-collapse">
34       <span class="sr-only">Toggle navigation</span>
35       <span class="icon-bar"></span>
36       <span class="icon-bar"></span>
37       <span class="icon-bar"></span>
38     </button>
39     <a class="navbar-brand" href="admin.php">Dziennik elektroniczny</a>
40   </div>
41   <ul class="nav navbar-right top-nav">
42     <li class="dropdown">
43       <a href="#" class="dropdown-toggle" data-toggle="dropdown"><i class="fa fa-user"></i> admin <b class="caret"></b></a>
44       <ul class="dropdown-menu">
45         <li>
46           <a href="show_profile.php"><i class="fa fa-fw fa-user"></i> Profile</a>
47         </li>
48         <li class="divider"></li>
49         <li>
50           <a href="includes/logout.php"><i class="fa fa-fw fa-power-off"></i> Log Out</a>
51         </li>
52       </ul>
53     </li>
54   </ul>
55   <div class="collapse navbar-collapse navbar-ex1-collapse">
56     <ul class="nav navbar-nav side-nav">
57       <li>
58         <a href="javascript:;" data-toggle="collapse" data-target="#uczen"><i class="fa fa-fw fa-arrows-v"></i> Uczniowie <i class="fa fa-fw fa-caret-down"></i></a>
59         <ul id="uczen" class="collapse">
60           <li>
61             <a href="show_student.php?source=add_student">Dodaj ucznia</a>
62           </li>
63           <li>
64             <a href="show_student.php">Lista uczniów</a>
65           </li>
66         </ul>
67       </li>
68       <li>
69         <a href="javascript:;" data-toggle="collapse" data-target="#nauczyciel"><i class="fa fa-fw fa-arrows-v"></i> Nauczyciele <i class="fa fa-fw fa-caret-down"></i></a>
70         <ul id="nauczyciel" class="collapse">
71           <li>
72             <a href="show_teacher.php?source=add_teacher">Dodaj nauczyciela</a>
73           </li>
74           <li>
75             <a href="show_teacher.php">Lista nauczycieli</a>
76           </li>
77         </ul>
78       </li>
79       <li>
80         <a href="show_class.php"><i class="fa fa-fw fa-dashboard"></i> Klasy</a>
81       </li>
82       <li>
83         <a href="show_subject.php"><i class="fa fa-fw fa-dashboard"></i> Przedmioty</a>
84       </li>
85       <li>
86         <a href="javascript:;" data-toggle="collapse" data-target="#plan"><i class="fa fa-fw fa-arrows-v"></i> Użytkownicy <i class="fa fa-fw fa-caret-down"></i></a>
87         <ul id="plan" class="collapse">
88           <li>
89             <a href="show_user.php?source=add_user">Dodaj użytkownika</a>
90           </li>
91           <li>
92             <a href="show_user.php">Lista użytkowników</a>
93           </li>
94         </ul>
95       </li>
96     </ul>
97   </div>
98 </nav>
99 <div id="page-wrapper">

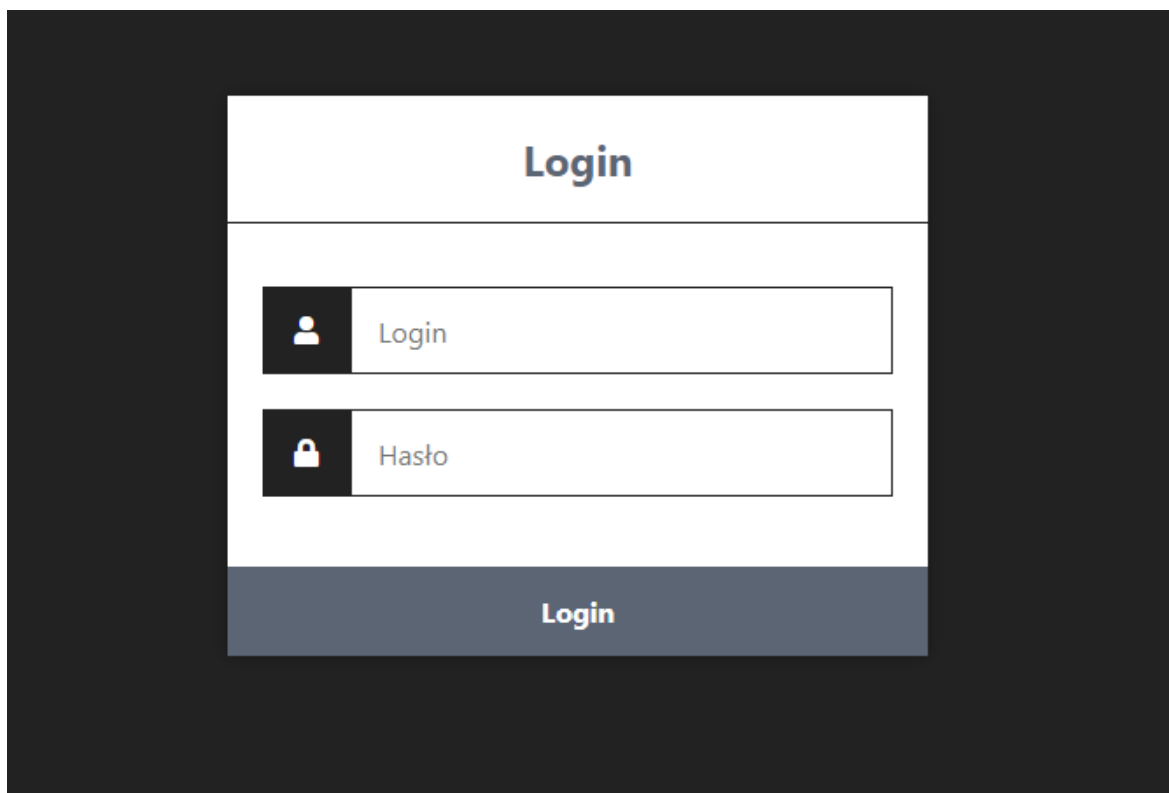
```

Jak widzimy element `` ("list item") jest używany do wskazywania punktu listy, zawartej w liście uporządkowanej. W tym przypadku mamy zawarte wszystkie elementy listy wraz z podpunktami, aby przykładowo po kliknięciu w „Uczniowie” rozwinęła nam się lista w przyciskami „Dodaj ucznia” oraz „Lista uczniów”. Taki zabieg pozwala zachować porządek na stronie oraz uporządkować dane, które widzi użytkownik, aby były bardziej przejrzyste i przyjazne. Powyższy kod oraz opisany przypadek dotyczy widoku administratora, który przekłada się na taki widok jaki możemy zobaczyć poniżej:



Jak widzimy po rozwinięciu zakładki „Uczniowie” widzimy dwie podkategorie, co dokładnie odpowiada temu, co możemy zobaczyć w kodzie HTML.

Poniżej przedstawiamy także kod odpowiedzialny za wygląd strony logowania:



```

body {
  background-color: #222;
}

.login {
  width: 400px;
  background-color: #ffffff;
  box-shadow: 0 0 9px 0 rgba(0, 0, 0, 0.3);
  margin: 100px auto;
}

.login h1 {
  text-align: center;
  color: #5b6574;
  font-size: 24px;
  padding: 20px 0 20px 0;
  border-bottom: 1px solid #222;
}

.login form {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
  padding-top: 20px;
}

.login form label {
  display: flex;
  justify-content: center;
  align-items: center;
  width: 50px;
  height: 50px;
  background-color: #222;
  color: #ffffff;
}

.login form input[type="password"], .login form input[type="text"] {
  width: 310px;
  height: 50px;
  border: 1px solid #222;
  margin-bottom: 20px;
  padding: 0 15px;
}

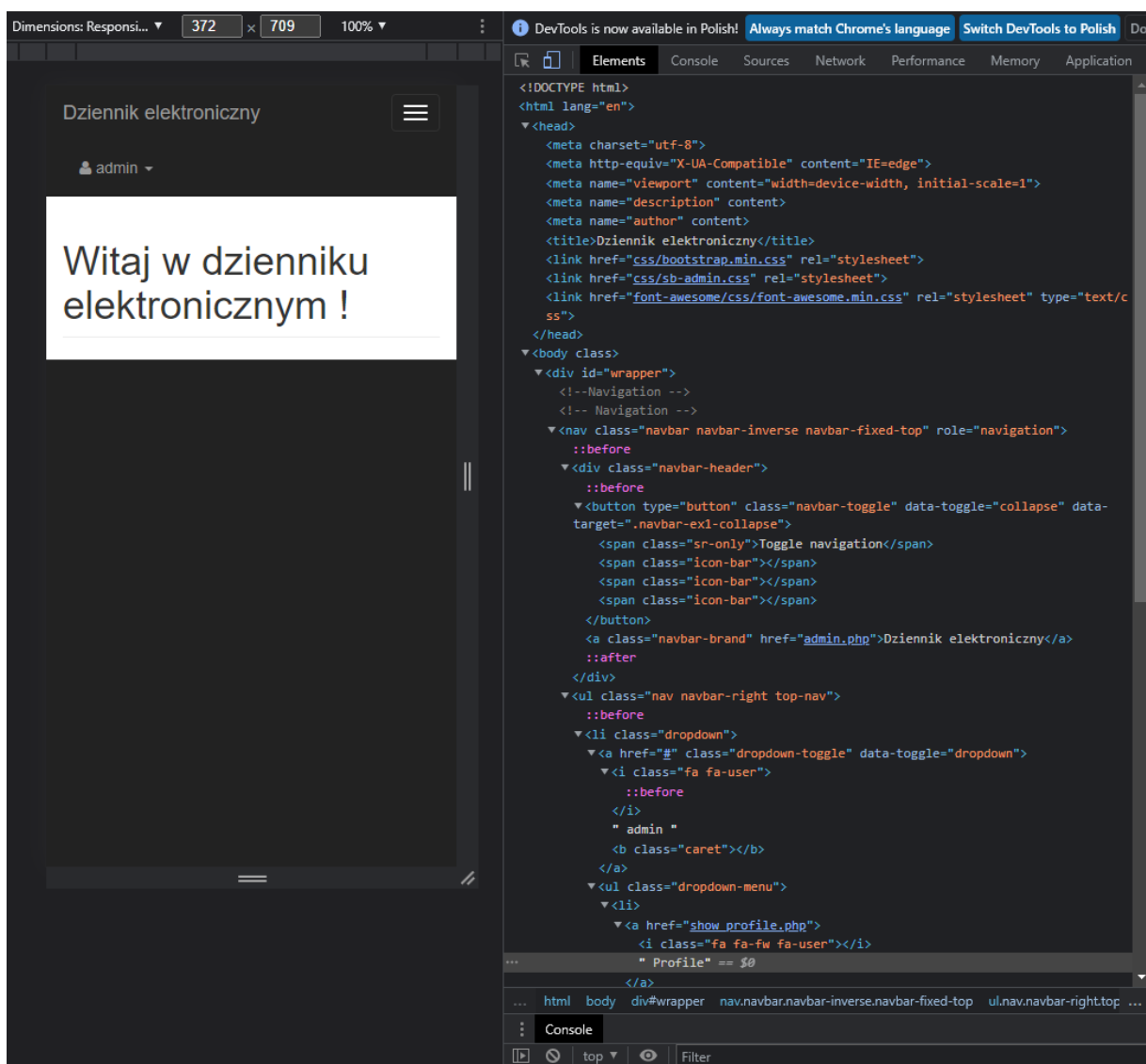
.login form input[type="submit"] {
  width: 100%;
  padding: 15px;
  margin-top: 20px;
  background-color: #5b6574;
  border: 0;
  cursor: pointer;
  font-weight: bold;
  color: #ffffff;
  transition: background-color 0.2s;
}

.login form input[type="submit"]:hover {
  background-color: #7a7d80;
  transition: background-color 0.2s;
}

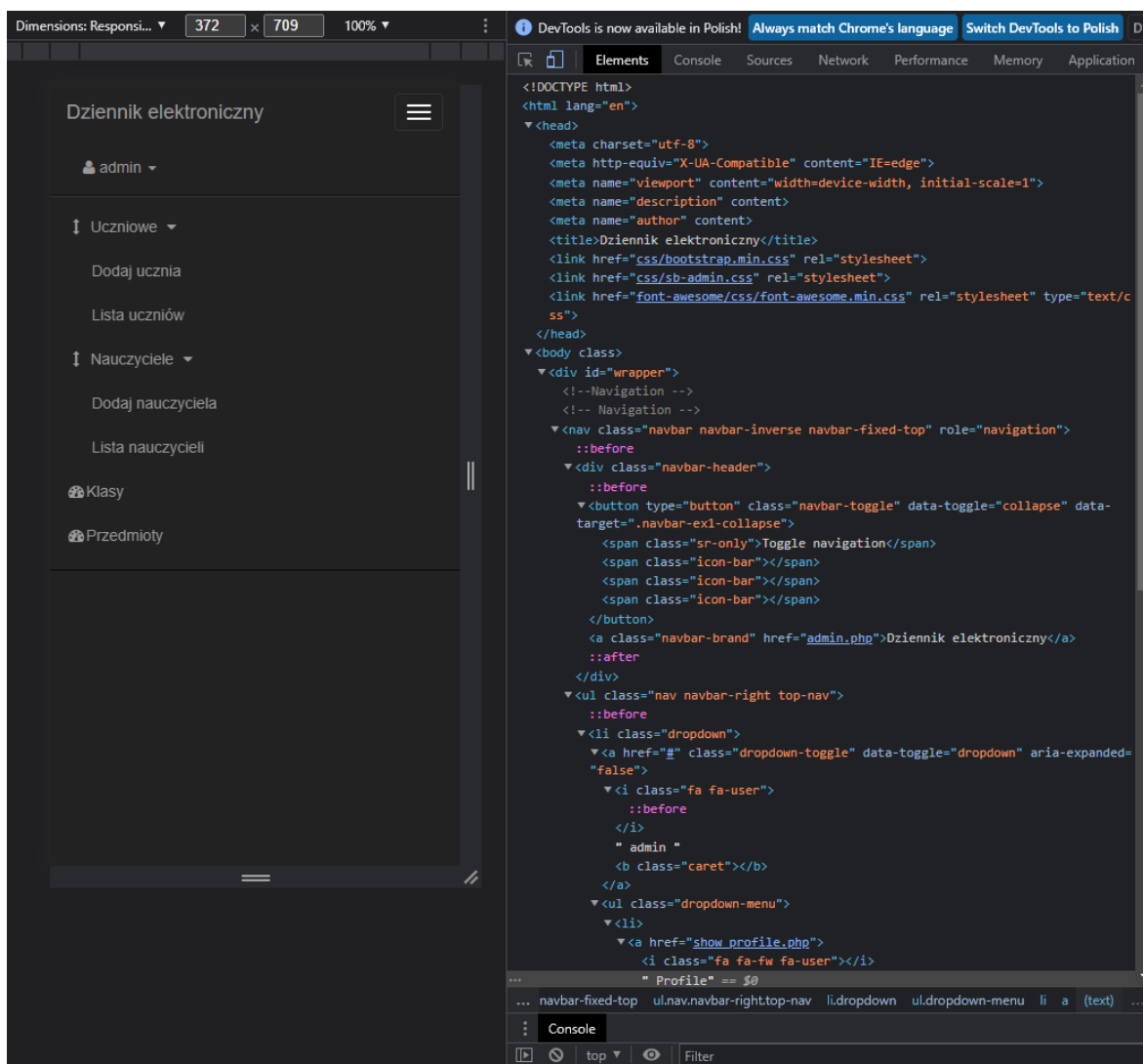
```



Dziennik elektroniczny jest także dostosowany do urządzeń mobilnych, aby umożliwić wszystkim użytkownikom swobodne korzystanie z dziennika również na smartfonie lub tablecie. Na poniższym zrzucie ekranu widzimy, w jaki sposób wygląda panel administratora z poziomu smartfona.



Zaprojektowana przez nas strona automatycznie wykrywa, w jakim trybie ma pracować na podstawie rozdzielczości ekranu, na której jest wyświetlana treść i dodaje przycisk „Menu”, czyli 3 pionowe kreski, po wciśnięciu których rozwija nam się dalsza część opcji, co obrazuje poniższy zrzut ekranu.



Celem dostosowanie strony internetowej do konkretnych rozdzielczości wykorzystaliśmy media query. Media query zbudowane jest z jej typu oraz przynajmniej jednego wyrażenia, dzięki któremu możliwe jest określenie zakresu działania kodu css, przez wykorzystanie takich wartości jak szerokość, wysokość lub kolor.

Przykładem wykorzystania media query może być poniższy fragment kodu, który zmienia parametry CSS dla maksymalnej rozdzielczości ekranu wynoszącej 480 pikseli.

```
4107 @media (max-device-width: 480px) and (orientation: landscape) {  
4108     .navbar-fixed-top .navbar-collapse,  
4109     .navbar-fixed-bottom .navbar-collapse {  
4110         max-height: 200px;  
4111     }  
4112 }  
4113 .container > .navbar-header,  
4114 .container-fluid > .navbar-header,  
4115 .container > .navbar-collapse,  
4116 .container-fluid > .navbar-collapse {  
4117     margin-right: -15px;  
4118     margin-left: -15px;  
4119 }  
4120 @media (min-width: 768px) {  
4121     .container > .navbar-header,  
4122     .container-fluid > .navbar-header,  
4123     .container > .navbar-collapse,  
4124     .container-fluid > .navbar-collapse {  
4125         margin-right: 0;  
4126         margin-left: 0;  
4127     }  
4128 }  
4129 .navbar-static-top {  
4130     z-index: 1000;  
4131     border-width: 0 0 1px;  
4132 }  
4133 @media (min-width: 768px) {  
4134     .navbar-static-top {  
4135         border-radius: 0;  
4136     }  
4137 }
```

W tym przypadku dodatkowym warunkiem do spełnienia jest tryb wyświetlania poziomy, jednak zastosowań samego media query w całym kodzie jest o wiele więcej. Pojawia się on w każdym miejscu, gdzie konieczne jest dostosowanie sposobu wyświetlania każdej rzeczy do trybu smartfona, tabletu, czy nawet telewizora, co oczywiście jak już wcześniej wspomniano jest uzależnione od rozdzielczości urządzenia, na którym wyświetlana jest dana treść.

C3. Efekty realizacji projektu

Strona została stworzona dla nauczycieli i uczniów, którzy mogą sprawdzić poszczególne dane. W przypadku dostępu przez nauczyciela, może on modyfikować i wprowadzać dane do których mają wgląd uczniowie. Nasz system jest czytelny i łatwy w obsłudze na czym nam zależało ze względu na korzystanie przez osoby w różnym przedziale wiekowym. Pierwsze logowanie do systemu odbywa się po wcześniejszej utworzeniu konta przez administratora strony.



C4. Użyteczność projektu

Projekt został stworzony dla użytku placówki szkolnej przez nauczycieli i uczniów, którzy wcześniej nie mieli możliwości korzystać z tego typu oprogramowania. Naszym celem było usprawnić pracę nauczycieli oraz umożliwić uczniom kontrolowanie swoich ocen. Jest to również duże usprawnienie podczas pracy zdalnej lub poza murami szkoły. Każdy z użytkowników przypisany do systemu może swobodnie w dowolnym momencie zalogować się do platformy i z niej korzystać. Administrator w dowolnym momencie ma prawo dokonać wszelkich zmian, zarządzać prawami użytkowników, dodawać nowych i nadawać im określone role.

C5. Autoewaluacja projektu

Jakub Fronczyk

Realizowany projekt pomógł mi utrwalić, uporządkować oraz zwiększyć moją wiedzę z zakresu web developmentu. Praktycznie zastosowanie języka PHP, wysyłanie połączeń i zapytań do bazy danych, dynamiczne wyświetlanie informacji z bazy danych. Rozwinąłem również wiedzę z zakresu HTML oraz CSS, użyłem podstawowych funkcji języka JavaScript. Jestem z siebie zadowolony oraz z mojej grupy, że udało nam się zrealizować ten projekt. Praca w grupie również poruszyła ważne aspekty funkcjonowania w zespole projektowym, zwiększyła umiejętności interpersonalne oraz komunikowania się. Wierzę, że dzięki temu projektowi będę w stanie dalej poszerzać swoją wiedzę w tworzeniu aplikacji internetowych i zająć się tym w przyszłości zawodowo.

Adrian Trojanowski

Realizacja projektu pozwoliła mi zasięgnąć wiedzy na temat języka programowania PHP oraz implementacji go w projekcie. Rozwinąłem także moją wiedzę na temat języka HTML oraz CSS. Zaczerpnąłem wiedzy na temat Frameworku Bootstrap oraz mogłem uczestniczyć w implementowaniu tej technologii. Podczas realizacji projektu mogłem również podszkolić swoje umiejętności w języku SQL. Zarządzanie bazą danych z pewnością jest bardzo przydatną umiejętnością. Również sama możliwość pracy w grupie oraz zarządzania zadaniami w czasie na pewno zwiększyła moje zdolności pracy w zespole oraz pozwoliła bardziej poznać samego siebie. Pomimo ciężkiej presji czasu oraz konieczności poszerzenia wiedzy z technologii, które nie były mi znane, czuję że praca nad



tym projektem znacznie poszerzyła moją wiedzę oraz umiejętności, które bez wątpienia zaowocują w przyszłości i będą przydatne w pracy.

Miłosz Fizyczak

Realizując projekt utrwaliłem oraz pozyskałem dodatkową wiedzę z zakresu języka HTML i CSS, poznałem również Framework o nazwie Bootstrap, dzięki któremu poznałem nowe funkcje oraz zastosowanie. Poszerzyłem swoją wiedzę z zakresu projektowania diagramów UML co pozwoliło mi spojrzeć na tworzoną stronę internetową pod kątem technicznym. Utrwaliłem i rozszerzyłem swoje wiadomości na temat języka SQL .

W projekcie wykorzystałem wiedzę nabytą podczas całego toku trwania studiów, organizowania pracy własnej jak i zespołowej. Umiejętności, które nabyłem podczas tworzenia projektu będą miły zastosowanie w mojej pracy zawodowej .

C6. Wykorzystane materiały i bibliografia związana z realizacją projektu

- <https://www.php.net/>
- <https://www.w3schools.com/php/>
- <https://phptherightway.com/>
- <https://www.udemy.com/>
- <https://www.tutorialspoint.com/php/index.htm/>

C7. Spis załączników

- Wzór ankiety

Pracę wraz z kodem źródłowym oraz wyeksportowaną bazą danych można zobaczyć korzystając z poniższego linku prowadzącego do dysku w usłudze Google:

<https://drive.google.com/drive/folders/18APyJopdH2Od-URG3vw5BTbHrj-4gaHe?usp=sharing>