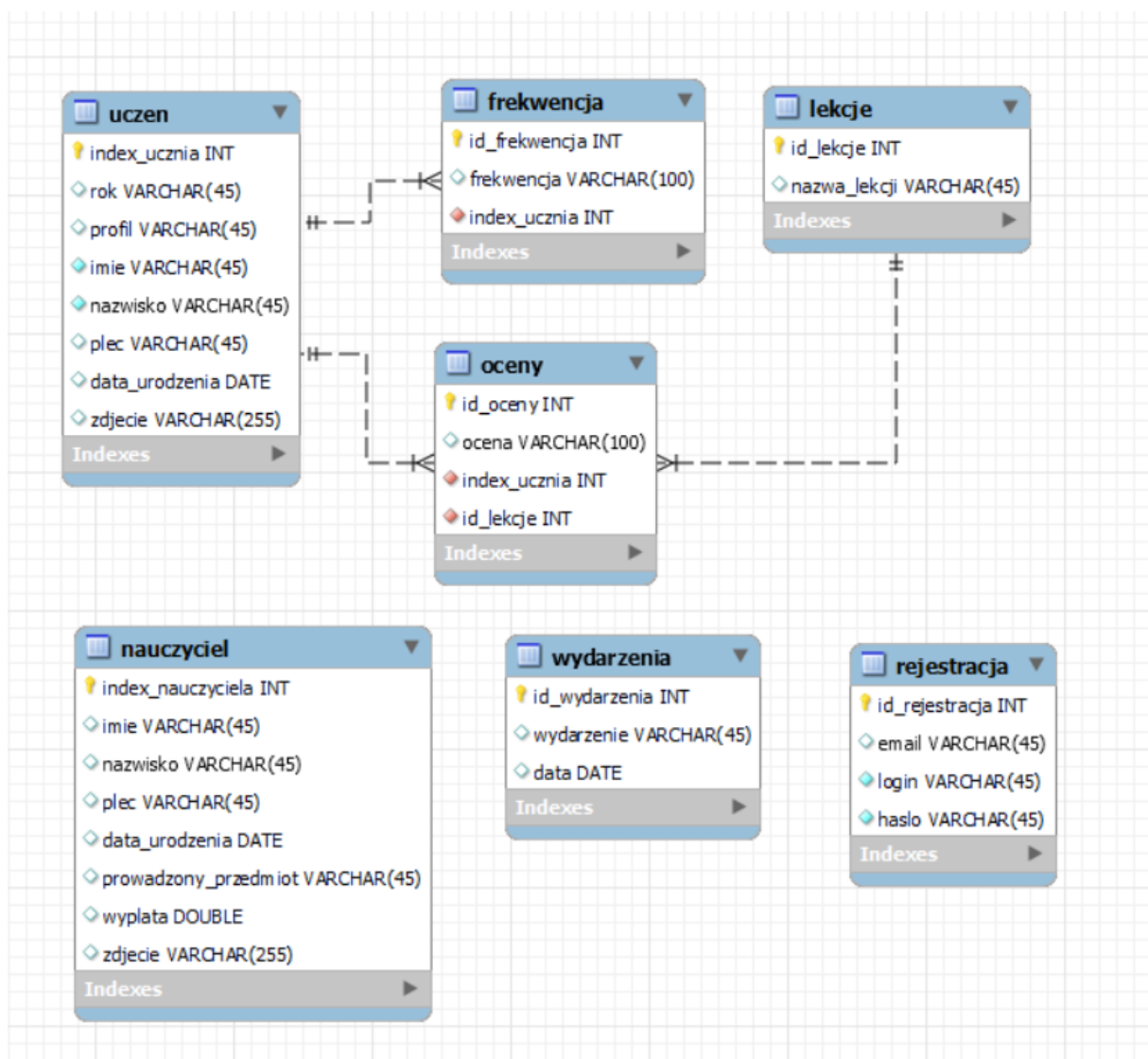


Projekt z przedmiotu **Programowanie Obiektowe**  
Temat projektu: **Aplikacja desktopowa w języku Java**  
Nazwa projektu: **Dziennik Szkolny**  
Autor: **Karol Bal 123630 Lab1**  
Data oddania: **18.01.2023**  
Prowadzący: **mgr inż. Wojciech Gałka**  
Rok akademicki: **2022/2023**

## 1. Wprowadzenie

- 1.1 Celem aplikacji jest zarządzanie dziennikiem szkolnym za pomocą analizy i wizualizacji danych w interfejsie graficznym użytkownika, jak i przedstawieniu możliwości dziennika szkolnego w interfejsie graficznym użytkownika. Aplikacja umożliwia dodawanie uczniów, nauczycieli oraz wydarzeń do bazy danych powiązanej z dziennikiem szkolnym. Ponadto umożliwia ona zarządzanie uczniami w konspekcie zarządzania ich frekwencją oraz możliwością wstawiania i edycji zdobytych ocen z poszczególnych przedmiotów. Aplikacja umożliwia także założenie własnego konta. Aplikacja umożliwia ponadto eksport danych wprowadzonych w tabeli do pliku pdf.
- 1.2 Wymagania: Komputer z systemem Windows i serwer z bazą danych.
- 1.3 Baza danych:

Diagram ERD



Baza danych składa się z 6 tabel: lekcje, oceny, frekwencja, uczen, rejestracja, wydarzenia

Tabela rejestracja zawiera dane na temat użytkownika niezbędne do zalogowania do aplikacji.

Tabela wydarzenia zawiera dane na temat zaplanowanych wydarzeń.

Tabela uczen zawiera dane na temat uczniów wprowadzonych do aplikacji, jest ona połączona relacją One to Many z tabelą oceny oraz tabelą frekwencja. Tabela do reprezentacji jaki uczeń dostał jaką ocenę z jakiego przedmiotu.

Tabela oceny to tabela która ma za zadanie łączyć tabele uczeń z tabelą lekcje tak aby magazynować informacje na temat ocen danego ucznia z danego przedmiotu.

Tabela lekcje jest to tabela zawierająca nazwy lekcji ma ona połączenie One to Many z tabelą oceny.

Tabela nauczyciel zawiera dane na temat nauczycieli wprowadzonych do aplikacji.

Tabela frekwencja zawiera dane potrzebne do wstawienia obecności danemu uczniowi z danego przedmiotu posiada ona połączenie Many to One z tabelą uczniowie.

## 2. Instalacja

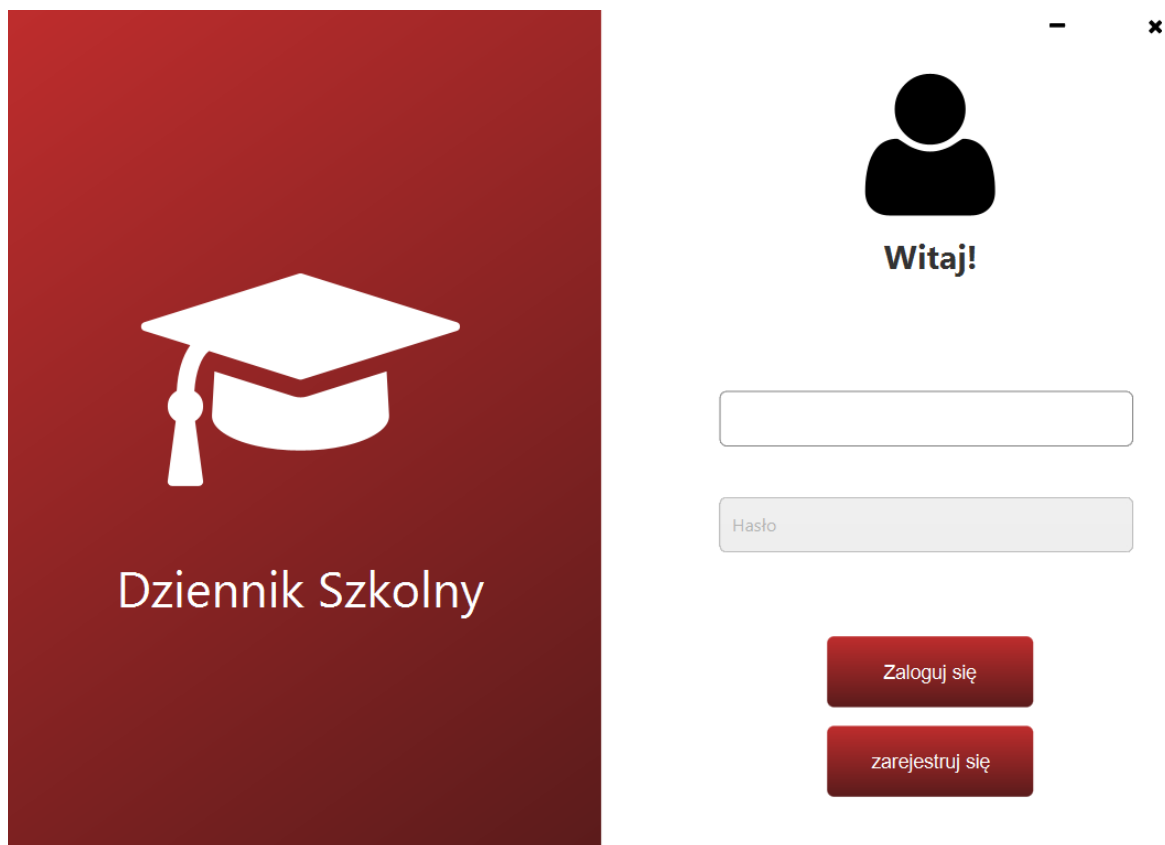
Do poprawnego działania aplikacji wymagany jest system Windows i Lokalny Server MySQL na tym serwerze musimy utworzyć bazę danych o nazwie projekt. Aby zainstalować aplikację klikamy w DziennikInstall.exe i podążamy za instrukcjami instalatora. Po instalacji aplikacji klikamy w DzeinnikRental.exe. Aby w aplikacji pojawiły się z=dane należy je zaimportować do bazy danych.

## 3. Interfejs

### 3.1 Interfejs ogólny

#### 3.1.1 Panel logowania

W oknie początkowym [Rys 1] możemy się zalogować jako użytkownik, lub przejść do okna rejestracji



[Rys 1] Okno Logowania

### 3.1.2 Panel rejestracji

W oknie rejestracji [Rys 2] możemy utworzyć nowe konto użytkownika

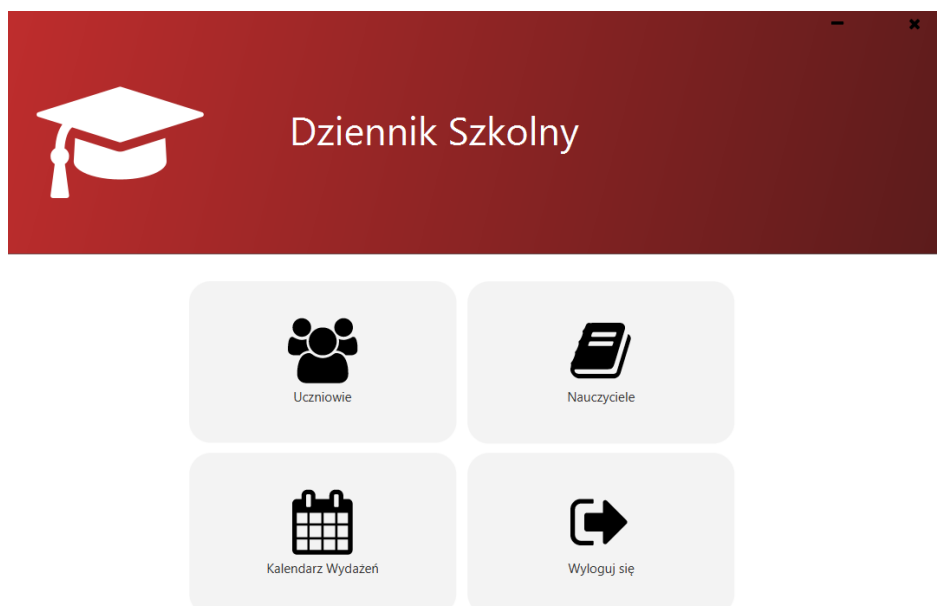


The registration window features a dark red header with a white graduation cap icon and the text "Dziennik Szkolny". Below the header, on a white background, is a registration form. At the top of the form is an icon of a person with a plus sign and the text "Zarejestruj się!". The form contains three input fields: "Adres email", "Nazwa użytkownika", and "Hasło". Below these fields are two buttons: "Zarejestruj się" (red) and "Masz już konto?" (red).

[Rys 2] Okno rejestracji

### 3.1.3 Menu główne

Menu główne posiada zestaw przycisków umożliwiających przejścia do dalszych interfejsów: panelu zarządzania uczniami, nauczycielami, wydarzeniami oraz przycisk umożliwiający wylogowanie się z aplikacji.



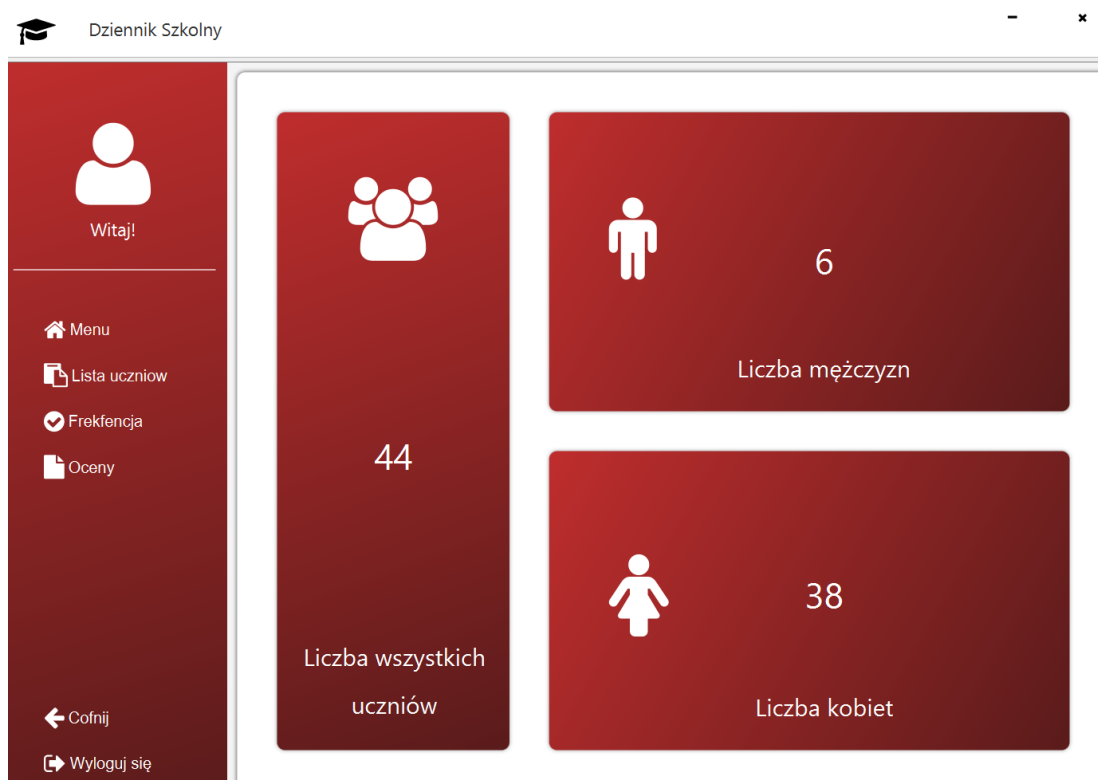
[Rys 3] Menu główne

## 3.2 Interfejs zarządzania uczniami

### 3.2.1 Panel powitalny zarządzania uczniami

Panel powitalny zarządzania uczniami posiada pasek nawigacyjny po lewej stronie, gdzie za pomocą przycisków możemy swobodnie przeglądać wybrane panele lub wylogować się z aplikacji oraz przycisk cofnij umożliwiający cofnięcie do panelu Menu główne.

W panelu powitalnym zarządzania uczniami [Rys 4] możemy sprawdzić ilu uczniów ogółem lub konkretnej płci zostało dodanych do aplikacji oraz za pomocą przycisków przejść do poszczególnych panelów aplikacji.



[Rys 4] Panel powitalny zarządzania uczniami

### 3.2.2 Panel Lista uczniów

W panelu Lista uczniów [Rys 5] możemy przeglądać dodanych uczniów, wprowadzić dane nowego ucznia, wyszukiwać uczniów po ich danych oraz wyeksportować widok tabeli do pliku pdf za pomocą przyciski Zapisz do pdf.



Witaj!

Menu

Lista uczniów

Frekwencja

Oceny

Cofnij

Wyloguj się

index Ucznia	Rok	Profil	Imię	Nazwisko	Płeć	Data urodzenia
1	II	biol-chem	Cynthia	Houndsom	Mężczyzna	2005-02-20
2	III	geograficzny	Aaren	Amorine	Mężczyzna	2011-12-20
3	II	humanistyczny	Nissie	Hatter	Kobieta	2005-02-20
4	III	biol-chem	Ced	Whelpton	Kobieta	2005-03-20
5	III	mat-inf	Karol	Gallety	Mężczyzna	2005-02-20
6	I	mat-inf	Kerwinn	Gudgion	Kobieta	2005-02-20
7	II	biol-chem	Madelina	Talboy	Kobieta	2009-11-20

1/6

1

2

3

4

5

6

index ucznia:

rok:

Profil:

Imię:

Nazwisko:

Płeć:

Data Urodzenia:

Dodaj

Usuń

edytuj

Wyczyść


Zaktualizuj

Zapisz do pdf

[Rys 5] Panel Lista uczniów


### 3.2.3 Panel Frekwencja


W panelu Frekwencja [Rys 6] możemy przeglądać oraz wprowadzać dane uczniów dotyczące ich procentowej obecności na zajęciach oraz wyeksportować widok tabeli do pliku pdf za pomocą przyciski Zapisz do pdf.





Witaj!


---


 Menu

 Lista uczniów

 Frekwencja

 Oceny

 Cofnij

 Wyloguj się

index ucznia	imie	nazwisko	matematyka(%)	język polski(%)	geografia(%)	informatyka(%)	wf(%)	biologia(%)
1	Cynthia	Houndsom	12.0	12.0	12.0	12.0	12.0	12.0
2	Aaren	Amorine	12.0	32.0	22.0	22.0	11.0	33.0
3	Nissie	Hatter	44.0	22.0	11.0	38.0	86.0	43.0
4	Ced	Whelpton	33.0	21.0	22.0	3.0	44.0	12.0
5	Karol	Gallety	33.0	22.0	11.0	22.0	33.0	21.0
6	Kerwinn	Gudgion	12.0	12.0	12.0	12.0	12.0	12.0
7	Madelina	Talboy	32.0	12.0				
8	Germana	Pelz						
9	Michaelina	Helstrom						
10	Chao	Lavalde						

< 1 2 3 4 5 6 >

1/6

index ucznia:

imie:

nazwisko:

matematyka:

język polski:

geografia:

informatyka:

wf:

biologia:

zapisz

wyczyszc


zaktualizuj

zapisz do pdf

[Rys 6] Panel Frekwencja


### 3.2.4 Panel Oceny


W panelu Oceny [Rys 7] znajduje się zestaw przycisków umożliwiających przejście do panelu wstawiania ocen.





Witaj!


---


 Menu


 Lista uczniów

 Frekwencja


 Oceny

 Cofnij


 Wyloguj się




Matematyka




Język Polski




Geografia



Informatyka



Wychowanie Fizyczne



Biologia



[Rys 7] Panel Oceny

### 3.2.5 Panel wstawiania ocen

W panelu wstawiania ocen [Rys 8] możemy przeglądać dane dotyczące uczniów i ich ocen z wybranego przedmiotu, edytować oceny oraz wyeksportować widok tabeli do pliku pdf za pomocą przyciski Zapisz do pdf.

**Dziennik Szkolny**

Witaj!

- Menu
- Lista uczniów
- Frekwencja
- Oceny

← Cofnij

→ Wyloguj się

index Ucznia	Imię	Nazwisko	Ocena	Ocena	Ocena	Ocena	Ocena	Ocena	Średnia
1	Cynthia	Houndsom	2.0	3.0	4.0	5.0	6.0	1.0	3,50
2	Aaren	Amorine							
3	Nissie	Hatter							
4	Ced	Whelpton							
5	Karol	Gallety							
6	Kerwinn	Gudgion							
7	Madelina	Talboy							
8	Germana	Pelz							
9	Michaelina	Helstrom							

1/2

index ucznia:

Imię:

Nazwisko:

Ocena 1:

Ocena 2:

Ocena 3:

Ocena 4:

Ocena 5:

Ocena 6:

Wyczyść

Zapisz

Zaktualizuj

Zapisz do pdf

[Rys 8] Panel wstawiania ocen

## 3.3 Interfejs zarządzania nauczycielami

### 3.3.1 Panel powitalny zarządzania nauczycielami

Panel powitalny zarządzania nauczycielami posiada pasek nawigacyjny po lewej stronie, gdzie za pomocą przycisków możemy swobodnie przeglądać wybrane panele lub wylogować się z aplikacji oraz przycisk cofnij umożliwiający cofnięcie do panelu Menu główne.

W panelu powitalnym zarządzania nauczycielami [Rys 9] możemy sprawdzić ile nauczycieli ogółem lub konkretnej płci zostało dodanych do aplikacji oraz za pomocą przycisków przejść do poszczególnych panelów aplikacji.



[Rys 9] Panel zarządzania nauczycielami

### 3.3.2 Panel Lista nauczycieli

W panelu Lista uczniów [Rys 10] możemy przeglądać dodanych uczniów, wprowadzić dane nowego ucznia, wyszukiwać uczniów po ich danych oraz wyeksportować widok tabeli do pliku pdf za pomocą przyciski Zapisz do pdf.



Witaj!

Menu

Lista nauczycieli

Cofnij

Wyloguj się

Wyszukaj

Index Nauczyciela	Imie	Nazwisko	Płeć	Data urodzenia	Prowadzony przedmiot	wypłata
1	Liane	Kilbourne	Męczyzna	1960-02-20	Geografia	2976.0
2	Kathleen	Klessmann	Kobieta	1971-12-20	Język Polski	2640.0
3	Timi	Bunning	Męczyzna	1990-02-20	Informatyka	2932.0
4	Lenard	Narracott	Męczyzna	1990-03-20	Geografia	2502.0
5	Kaitlynn	Orwell	Kobieta	1971-02-20	Wychowanie Fizyczne	2478.0
6	Xenia	Aven	Kobieta	1989-02-20	Wychowanie Fizyczne	2796.0
7	Ernaline	Currier	Męczyzna	1971-11-20	Matematyka	2453.0

1/3

Index Nauczyciela:

Imie:

Nazwisko:

Płeć:

Data Urodzenia:

Prowadzony Przedmiot:

Wypłata

Dodaj

Wyczyść

Usuń

Zaktualizuj

edytuj

Zapisz do pdf

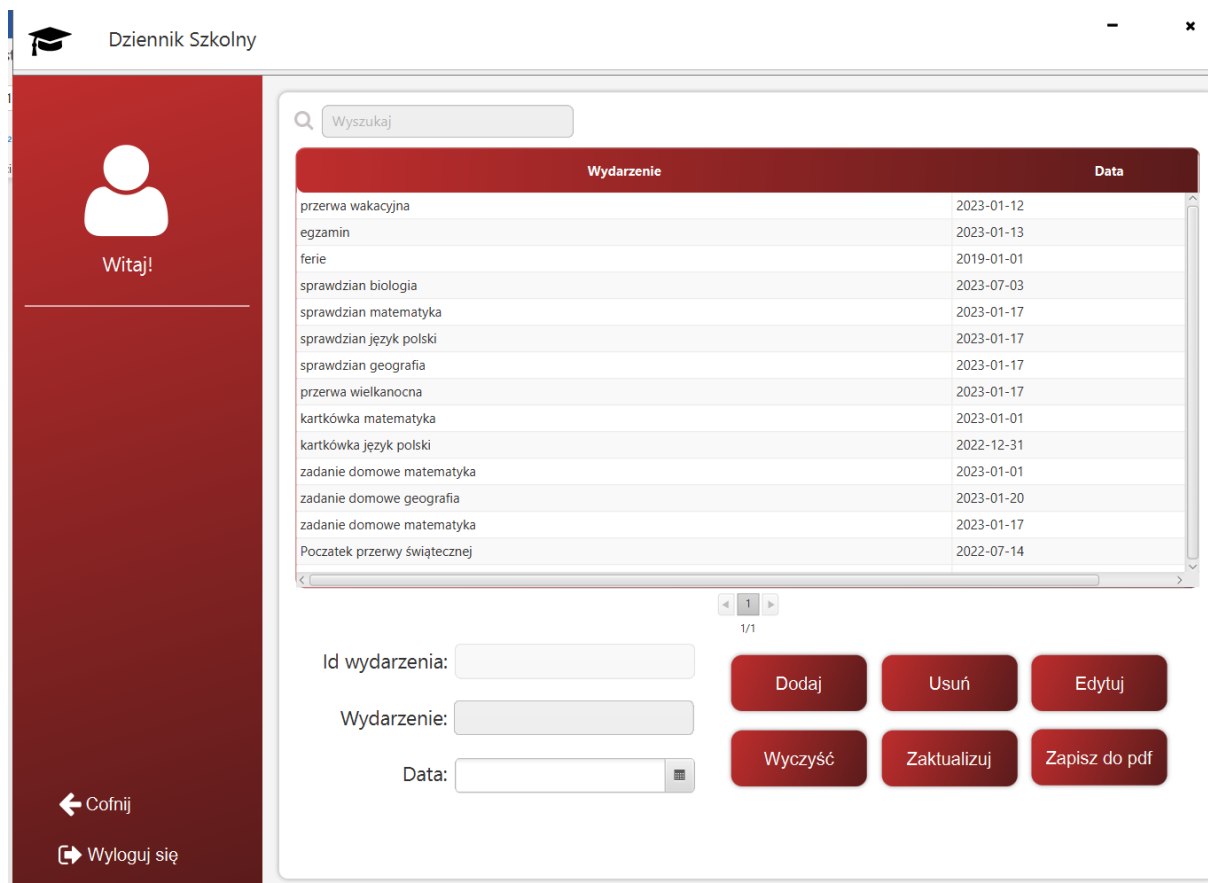
[Rys 10] Panel Lista nauczycieli

### 3.4 Interfejs zarządzania wydarzeniami

#### 3.4.1 Panel zarządzania wydarzeniami

Panel powitalny zarządzania wydarzeniami posiada pasek nawigacyjny po lewej stronie, gdzie za pomocą przycisku wyloguj możemy wylogować się z aplikacji oraz przycisk cofnij umożliwiający cofnięcie do panelu Menu główne.

W panelu zarządzania wydarzeniami [Rys 11] możemy przeglądać dodane wydarzenia, wprowadzić informacje o nowym wydarzeniu, wyszukiwać wydarzenie po jego danych oraz wyeksportować widok tabeli do pliku pdf za pomocą przyciski Zapisz do pdf.



[Rys 11] Panel zarządzania wydarzeniami

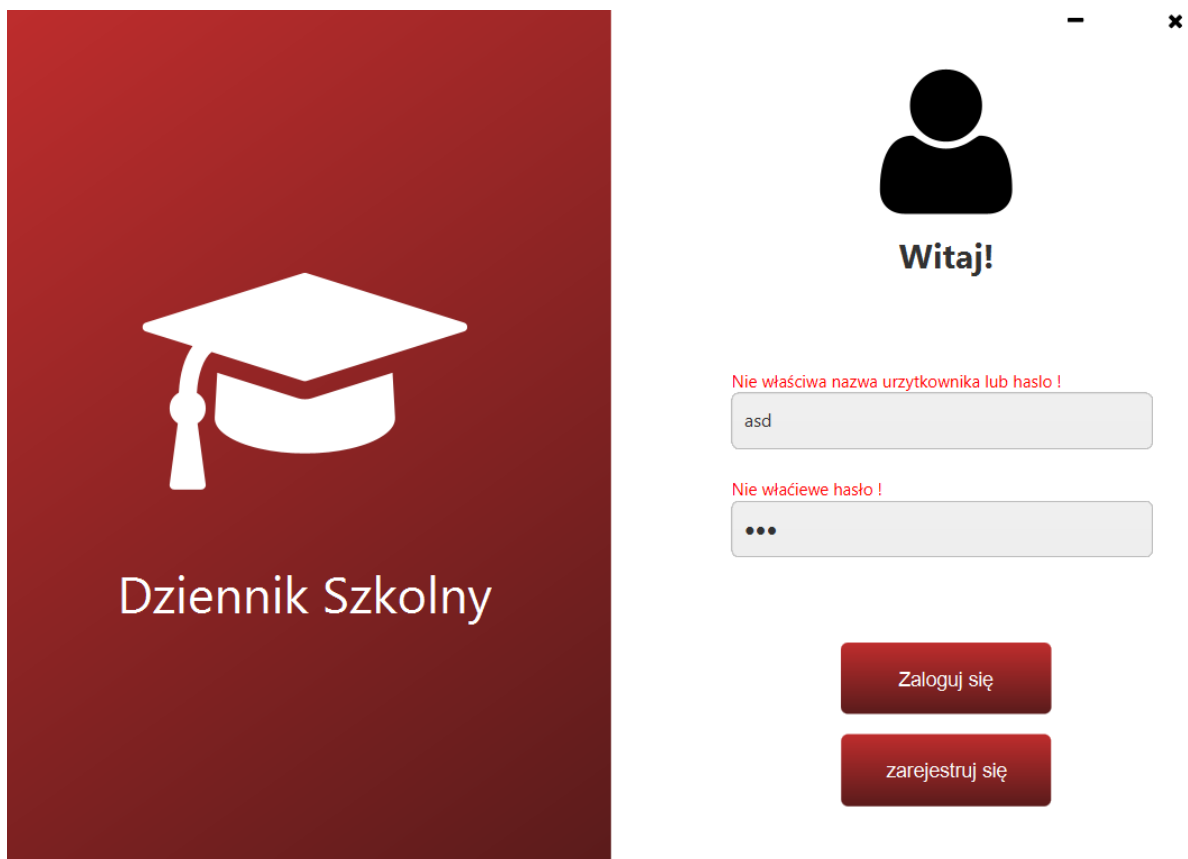
Pozostałe elementy interfejsów zostały przedstawione i opisane w punkcie 4. Funkcjonalność

## 4. Funkcjonalność

### 4.1 Lista funkcjonalności ogólnych

#### 4.1.1 Logowanie [Rys 1]

Okno logowania umożliwia zalogowanie się do panelu użytkownika. Proces logowania polega na łączeniu z bazą danych i sprawdzeniu czy podane przez użytkownika dane faktycznie istnieją w bazie danych, jeśli nie, zostanie przedstawiony odpowiedni komunikat [Rys 12] o błędnym loginie lub hasle. Natomiast jeśli dane użytkownika są poprawne zostanie on przełączony na panel menu główne.



[Rys 12] Komunikat o błędnych danych podczas logowania

#### 4.1.2 Rejestracja [Rys 2]

Okno rejestracji umożliwia utworzenie nowego konta użytkownika poprzez wpisanie wszystkich odpowiednich danych oraz kliknięcie przycisku zarejestruj. W przypadku podania nieprawidłowych danych zostaną wyświetlone komunikaty o wprowadzeniu nie poprawnych danych[Rys 13], natomiast jeśli proces rejestracji przebiegnie pomyślnie zostanie wyświetlony komunikat [Rys 14] o pomyślnej rejestracji, po czym użytkownik zostanie dodany do bazy danych.



## Zarejestruj się!

Nie prawidłowy adres email !

karol

Nie prawidłowa nazwa użytkownika !

123

Nie prawidłowe hasło !

...

Zarejestruj się

Masz już konto?



[Rys 13] Komunikaty o błędnych danych podczas rejestracji



**Zarejestruj się!**

karooooo!@asdfgh

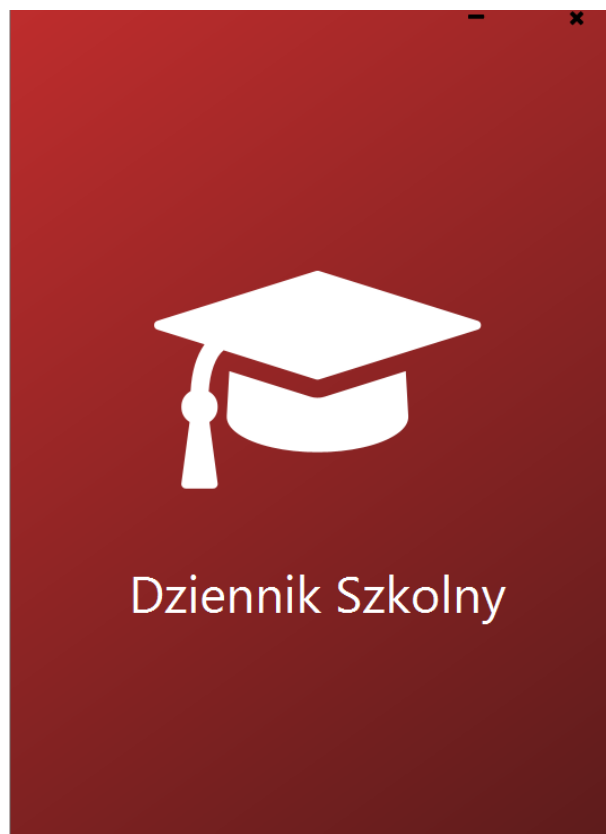
karool

.....

Pomyślnie Zarejestrowano !

Zarejestruj się

Masz już konto?



[Rys 14] Komunikat o pomyślnym zarejestrowaniu

#### 4.1.3 Wylogowanie, „x” i „-”

Przyciski wyloguj, „x” i „-” są wspólne dla panelu wszystkich paneli. Po naciśnięciu przycisku wyloguj, aplikacja zamyka dany panel i ponownie wyświetla panel logowanie. Przycisk „x” wywołuje kompletne zamknięcie aplikacji. Natomiast przycisk „-” jej zminimalizowanie.


### 4.2 Lista funkcjonalności panelu zarządzania uczniami

#### 4.2.1 Przeglądanie uczniów

Panel lista uczniów umożliwia użytkownikowi przeglądanie wprowadzonych do bazy danych uczniów

#### 4.2.2 Wyszukiwanie uczniów

Panel lista uczniów umożliwia użytkownikowi wyszukiwanie ucznia po danych wprowadzonych do bazy danych, podczas wpisywania danej poszukiwanego ucznia widok uczniów dynamicznie aktualizuje uczniów w tabeli na podstawie wpisanej danej [Rys 15]



Witaj!

- Menu
- Lista uczniów
- Frekwencja
- Oceny

[← Cofnij](#)

[Wyloguj się](#)

index Ucznia	Rok	Profil	Imię	Nazwisko	Płeć	Data urodzenia
32	II	mat-inf	karol	asfgdsh	Kobieta	2022-12-23
45	I	mat-inf	karol	bal	Mężczyzna	2023-01-05

index ucznia:   
rok:   
Profil:   
Imię:   
Nazwisko:   
Płeć:   
Data Urodzenia:

Dodaj

Usuń

edytuj

Wyczyść

Zaktualizuj

Zapisz do pdf

[Rys 15] Wyszukiwanie uczniów

#### 4.2.3 Wprowadzanie danych ucznia

Podczas wprowadzania ucznia do bazy danych wszystkie pola muszą zostać poprawnie uzupełnione i nie mogą pozostać puste w przeciwnym razie zostanie wyświetlony stosowny komunikat Rys[16]. Natomiast po wprowadzeniu danych w poprawny sposób i następnym naciśnięciu przycisku dodaj wyświetlony zostanie stosowny komunikat Rys[17] a uczeń zostanie dodany do bazy.

#### 4.2.4 Usuwanie danych ucznia

Podczas usuwania ucznia z bazy danych wszystkie pola muszą zostać poprawnie uzupełnione i nie mogą pozostać puste w przeciwnym razie zostanie wyświetlony stosowny komunikat Rys[16]. Natomiast po wprowadzeniu danych ucznia w poprawny sposób i następnym naciśnięciu przycisku usuń wyświetlony zostanie stosowny komunikat Rys[17] a uczeń zostanie usunięty z bazy.



Wyszukaj

index Ucznia	Rok	Profil	Imię	Nazwisko	Płeć	Data urodzenia
1	II	biol-chem	aa	aa	Kobieta	2022-12-01
2	II	mat-inf	ggt	oke	Kobieta	2022-12-08
3	II	mat-inf	ggt	oke	Kobieta	2022-12-08
4	II	biol-chem	dsaffdhg	sdghd	Kobieta	2022-12-10
6	III	humanistyczny	twoja	sdfgh	Mężczyzna	2025-12-13
7	I	mat-inf	agag	sadgasgd	Kobieta	2022-12-16
8	I	biol-chem	asghd	adfhj	Kobieta	2022-12-01

index ucznia:

Niewłaściwy index !

rok:

niewłaściwy rok !

Profil:

Niewłaściwy profil !

Imię:

Niewłaściwe imię !

Nazwisko:

Niewłaściwe Nazwisko !

Płeć:

Niewłaściwe płeć !

Data Urodzenia:

Niewłaściwa data urodzenia !

Dodaj

Usuń

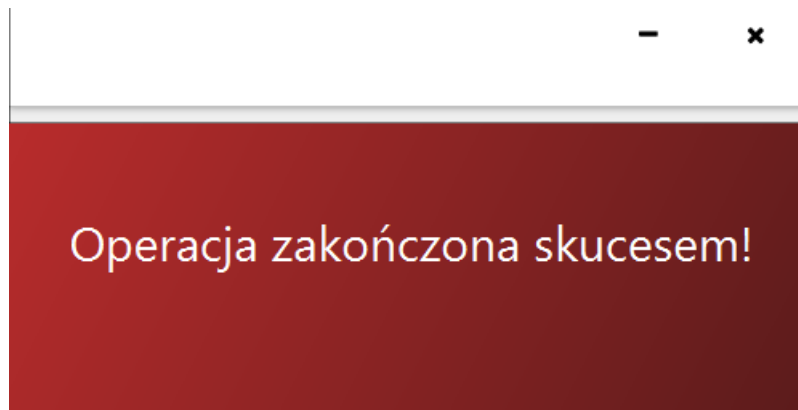
edytuj

Wyczyść

Zaktualizuj

Zapisz do pdf

[Rys 16] Walidacja wprowadzania ucznia



[Rys 17] Komunikat po poprawnym przeprowadzeniu operacji

#### 4.2.5 Edycja danych ucznia

Użytkownik ma możliwość edycji danych ucznia po naciśnięciu przycisku „Edytuj” [Rys 5], po naciśnięciu przycisku jeżeli dane zostały wprowadzone prawidłowo dane zostaną zmienione oraz wyświetli się komunikat [Rys 17], natomiast jeżeli dane zostały wprowadzone w sposób nie prawidłowy zostanie wyświetlona informacja o nie prawidłowych danych [Rys 16] w zależności jaka dana została źle wprowadzona.

#### 4.2.6 Zapis danych ucznia do pdf

Użytkownik ma możliwość zapisu danych ucznia po naciśnięciu przycisku „Zapisz do pdf” [Rys 5], po naciśnięciu przycisku dane zostaną zapisane do pdf [Rys 18] oraz wyświetli się komunikat [Rys 17].

#### 4.2.7 Wyczyszczenie pul do wprowadzania danych

Użytkownik ma możliwość wyczyszczenia pul za pomocą wciśnięcia przycisku „Wyczyść” [Rys 5], po naciśnięciu przycisku nastąpi wyczyszczenie pul oraz wyświetli się komunikat [Rys 17].

#### 4.2.8 Zaktualizowanie tabeli

Użytkownik ma możliwość zaktualizowania tabeli za pomocą wciśnięcia przycisku „Zaktualizuj” [Rys 5], po naciśnięciu przycisku nastąpi zaktualizowanie tabeli.

1	aa	aa	Kobieta	2022-12-01	biol-chem	II
2	ggt	oke	Kobieta	2022-12-08	mat-inf	II
3	ggt	oke	Kobieta	2022-12-08	mat-inf	II
4	dsaffdhg	sdghd	Kobieta	2022-12-10	biol-chem	II
6	twoja	sdfgh	Mczyzna	2025-12-13	humanisty czny	III
7	agag	sadgasgd	Kobieta	2022-12-16	mat-inf	I
8	asghd	adfjh	Kobieta	2022-12-01	biol-chem	I
9	fsgdhfj	agsdhj	Kobieta	2022-12-16	mat-inf	I
10	afsgdhfg	dfsghfg	Kobieta	2022-12-09	biol-chem	I
11	aa	aa	Kobieta	2022-12-01	biol-chem	I
12	ggt	oke	Kobieta	2022-12-08	biol-chem	II
13	dafsgfhg	fdghfjgh	Kobieta	2022-12-01	biol-chem	I
14	rstdffg	fgxdhfjgk	Kobieta	2022-12-15	biol-chem	I
15	hhhh	ddd	Kobieta	2022-12-02	biol-chem	II
16	adsfs	afg	Kobieta	2022-12-08	mat-inf	II
17	afdsdg	fdsgdghf	Kobieta	2022-12-07	biol-chem	I
18	asdf	afds	Kobieta	2022-12-24	biol-chem	I
19	sdfg	dfdf	Kobieta	2022-12-1	biol-chem	II


[Rys 18] Dane ucznia zapisane w pdf

#### 4.2.9 Przeglądanie danych w panelu frekwencja

Panel frekwencja umożliwia użytkownikowi przeglądanie wprowadzonych do bazy danych danych o frekwencji uczniów

#### 4.2.10 Wprowadzanie danych w panelu frekwencja

Podczas wprowadzania informacji o frekwencji do bazy danych wszystkie pola muszą zostać poprawnie uzupełnione w przeciwnym razie zostanie wyświetlony stosowny komunikat Rys[19]. Natomiast po wprowadzeniu nauczyciela w poprawny sposób i następnym naciśnięciu przycisku dodaj wyświetlony zostanie stosowny komunikat Rys[17] a frekwencja zostanie zaktualizowana w bazie.



Witaj!

- Menu
- Lista uczniów
- Frekwencja
- Oceny

Cofnij

Wyloguj się

index ucznia	imie	nazwisko	matematyka(%)	język polski(%)	geografia(%)	informatyka(%)	wf(%)	biologia(%)
4	dsaffdhg	sdghd	0	0	0	0	0	0
18	asdf	afds	12	17	12	12	21	32
19	sdfg	dfdf	0	100	0	0	0	0
22	cvbn	fsddgfh	21	2	22	23	24	21
23	asdsffd	asfdd	0	0	0	0	0	0
24	asfsg	afdsd	0	0	0	0	0	0
25	dsfg	fdsgfd	0	0	2	0	0	0
26	dsfg	fdsgfd	0	0	0	0	0	0
27	dsfg	fdsgfd	0	0	0	0	0	0
28	dsfg	fdsgfd	0	0	0	0	0	0
29	dsfg	fdsgfd	0	0	0	0	0	0
30	dsfg	fdsgfd	0	0	0	0	0	0

index ucznia:

imie:

nazwisko:

matematyka:

język polski:

geografia:

informatyka:

wf:

biologia:

zapisz

wyczysc

zaktualizuj

zapisz do pdf

Nie właściwy procent obecności!

[Rys 20] Walidacja danych frekwencja

#### 4.2.11 Zapis danych do pdf w panelu frekwencja

Użytkownik ma możliwość zapisu frekwencji ucznia po naciśnięciu przycisku „Zapisz do pdf” [Rys 6], po naciśnięciu przycisku dane zostaną zapisane do pdf[Rys 21] oraz wyświetli się komunikat [Rys 17].

t > projekt		Przeszukaj: projekt							
Nazwa	Data modyfikacji	Typ							
.idea	18.01.2023 18:04	Folder plik							
.mvn	10.01.2023 21:11	Folder plik							
out	18.01.2023 11:44	Folder plik							
src	10.01.2023 21:11	Folder plik							
target	18.01.2023 11:56	Folder plik							
.gitignore	02.12.2022 15:36	Dokument							
frekwencja.pdf	18.01.2023 18:04	Microsoft I							
mvnw	02.12.2022 15:36	Plik							
mvnw.cmd	02.12.2022 15:36	Skrypt pole							
nauczyciele.pdf	18.01.2023 18:02	Microsoft I							
oceny.pdf	16.01.2023 17:05	Microsoft I							
pom.xml	18.01.2023 11:59	Dokument							
projekt.iml	18.01.2023 12:39	Plik IML							
uczniowie.pdf	18.01.2023 11:12	Microsoft I							

4	dsaffdhg	sdghd	0	0	0	0	0	0
18	asdf	afds	12	17	12	12	21	32
19	sdfg	dfdf	0	100	0	0	0	0
22	cvbn	fsddgh	21	2	22	23	24	21
23	asdsff	asfdd	1	2	3	4	5	6
24	asfsg	afdsd	0	0	0	0	0	0
25	dsfg	fdsgfd	0	0	2	0	0	0
26	dsfg	fdsgfd	0	0	0	0	0	0
27	dsfg	fdsgfd	0	0	0	0	0	0
28	dsfg	fdsgfd	0	0	0	0	0	0
29	dsfg	fdsgfd	0	0	0	0	0	0
30	dsfg	fdsgfd	0	0	0	0	0	0
31	dsfg	fdsgfd	0	0	0	0	0	0
32	karol	asfgdsh	12	13	0	0	0	0
33	francze	asfgdsh	0	0	0	0	0	0
34	francze	asfgdsh	0	0	0	0	0	0
35	francze	asfgdsh	0	0	0	0	0	0
36	francze	asfgdsh	0	0	0	0	0	0
37	francze	asfgdsh	56	0	0	0	0	0
38	francze	asfgdsh	0	0	0	0	0	0
39	francze	asfgdsh	0	0	0	0	0	0

[Rys 21] Zapis do pdf frekwencja

#### 4.2.12 Wyczyszczenie pul do wprowadzania danych

Użytkownik ma możliwość wyczyszczenia pul za pomocą wciśnięcia przycisku „Wyczyść” [Rys 6], po naciśnięciu przycisku nastąpi wyczyszczenie pul oraz wyświetli się komunikat [Rys 17].

#### 4.2.13 Zaktualizowanie tabeli

Użytkownik ma możliwość zaktualizowania tabeli za pomocą wciśnięcia przycisku „Zaktualizuj” [Rys 6], po naciśnięciu przycisku nastąpi zaktualizowanie tabeli.

#### 4.2.14 Dodawanie ocen dla poszczególnych uczniów z poszczególnych przedmiotów

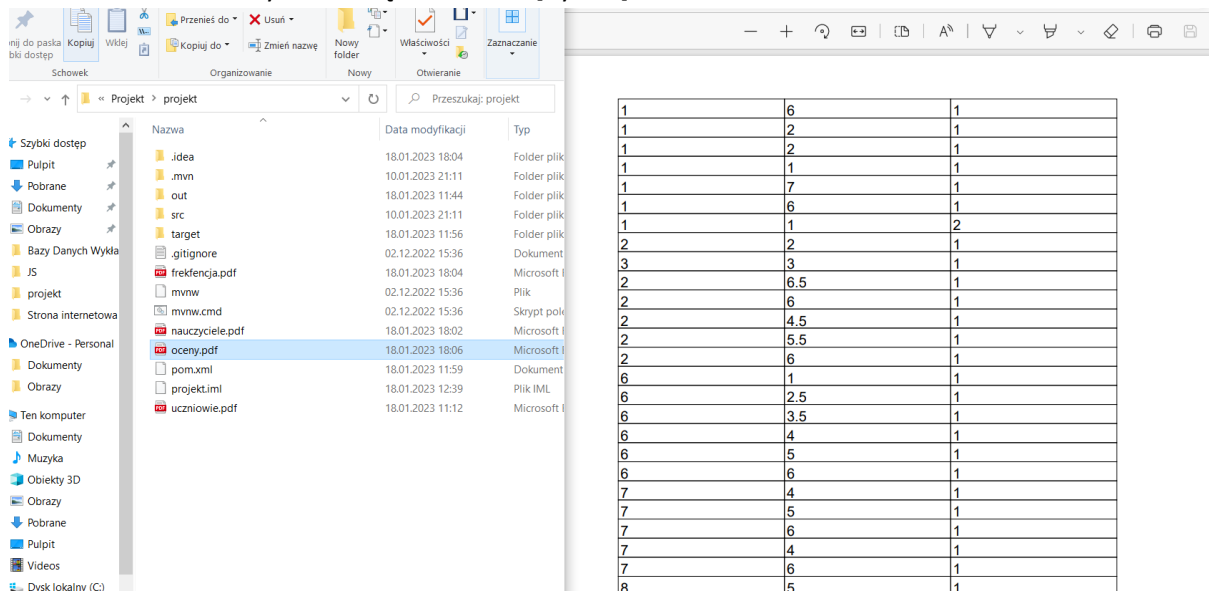
Użytkownik ma możliwość dodawania ocen dla poszczególnych uczniów z wybranego przedmioty po przejściu z panelu oceny [Rys 7] do panelu wstawiania ocen Rys[8]. Po wprowadzeniu oceny w dane pole i wciśnięciu przycisku „Zapisz” Rys[7] następuje zapisane danych podanych przez użytkownika w bazie danych.

#### 4.2.15 Wyczyszczenie pul do wprowadzania danych

Użytkownik ma możliwość wyczyszczenia pul za pomocą wciśnięcia przycisku „Wyczyść” [Rys 7], po naciśnięciu przycisku nastąpi wyczyszczenie pul oraz wyświetli się komunikat [Rys 17].

#### 4.2.16 Zapis danych do pdf w panelu wstawiania ocen

Użytkownik ma możliwość zapisu ocen ucznia po naciśnięciu przycisku „Zapisz do pdf” [Rys 7], po naciśnięciu przycisku dane zostaną zapisane do pdf [Rys] oraz wyświetli się komunikat [Rys 17].



[Rys 22] Zapis do pdf oceny

### 4.3 Lista funkcjonalności panelu zarządzania nauczycielami


#### 4.3.1 Przeglądanie danych nauczyciela

Panel lista nauczycieli umożliwia użytkownikowi przeglądanie wprowadzonych do bazy danych uczniów

#### 4.3.2 Wyszukiwania nauczyciela

Panel lista nauczycieli umożliwia użytkownikowi wyszukiwanie nauczyciela po danych wprowadzonych do bazy danych, podczas wpisywania danych poszukiwanego nauczyciela widok nauczycieli dynamicznie aktualizuje nauczycieli w tabeli na podstawie wpisanej danej [Rys 19]





Witaj!

Menu

Lista nauczycieli

Cofnij

Wyloguj się

fra

index Nauczyciela	Imię	Nazwisko	Płeć	Data urodzenia	Prowadzony przedmiot	wypłata
40	frank	nowak	Mężczyzna	2023-01-12	Geografia	12345.0

Index Nauczyciela: 15

Imię: Karol

Nazwisko: Bal

Płeć: Kobieta

Data Urodzenia: 27.01.2023

Prowadzony Przedmiot: Geografia

Wypłata: 1234567.0

Dodaj

Wyczyść

Usuń

Zaktualizuj


edytuj

Zapisz do pdf

[Rys 23] Wyszukiwanie nauczyciela

#### 4.3.3 Wprowadzanie nauczyciela

Podczas wprowadzania nauczyciela do bazy danych wszystkie pola muszą zostać poprawnie uzupełnione i nie mogą pozostać puste w przeciwnym razie zostanie wyświetlony stosowny komunikat Rys[24]. Natomiast po wprowadzeniu nauczyciela w poprawny sposób i następnym naciśnięciu przycisku dodaj wyświetlony zostanie stosowny komunikat Rys[25] a nauczyciel zostanie dodany do bazy.



Witaj!

Menu

Lista nauczycieli

Cofnij

Wyloguj się

index Nauczyciela	Imie	Nazwisko	Płeć	Data urodzenia	Prowadzony przedmiot	wypłata
1	dhj	dfsgh	Mężczyzna	2022-12-22	Geografia	1234567.0
2	ddfsghdf	dgshfdj	Kobieta	2022-12-07	Informatyka	2134567.0
3	afgdshddsgfgg	sagdsfhffghj	Mężczyzna	2022-12-15	Informatyka	1245444.0
4	bdsghdfj	dsgathdj	Kobieta	2022-12-28	Wychowanie Fizyczne	67890.0
5	bdsghdfj	dsgathdj	Kobieta	2022-12-28	Wychowanie Fizyczne	67890.0
6	bdsghdfj	dsgathdj	Kobieta	2022-12-28	Wychowanie Fizyczne	67890.0
7	bdsghdfj	dsgathdj	Kobieta	2022-12-28	Wychowanie Fizyczne	67890.0

Błędnie wprowadzony index !

Index Nauczyciela:

Błędnie wprowadzone imię !

Imię:

Błędnie wprowadzone nazwisko !

Nazwisko:

Błędnie wprowadzona płeć !

Płeć:

Błędnie wprowadzona data urodzenia !

Data Urodzenia:

Błędnie wprowadzony przedmiot !

Prowadzony Przedmiot:

Błędnie wprowadzona wypłata !

Wypłata:

Dodaj

Wyczyść

Usuń

Zaktualizuj

edytuj

Zapisz do pdf

[Rys 24] Walidacja wprowadzania nauczyciela

#### 4.3.4 Edytowanie danych nauczyciela

Użytkownik ma możliwość edycji danych nauczyciela po naciśnięciu przycisku „Edytuj” [Rys 10], po naciśnięciu przycisku jeżeli dane zostały wprowadzone prawidłowo dane zostaną zmienione oraz wyświetli się komunikat [Rys 17], natomiast jeżeli dane zostały wprowadzone w sposób nie prawidłowy zostanie wyświetlona informacja o nie prawidłowych danych [Rys 24] w zależności jaka dana została źle wprowadzona.

#### 4.3.5 Usuwanie danych nauczyciela

Podczas usuwania nauczyciela z bazy danych wszystkie pola muszą zostać poprawnie uzupełnione i nie mogą pozostać puste w przeciwnym razie zostanie wyświetlony stosowny komunikat Rys[24]. Natomiast po wprowadzeniu danych nauczyciela w poprawny sposób i następnym naciśnięciu przycisku usuń wyświetlony zostanie stosowny komunikat Rys[17] a nauczyciel zostanie usunięty z bazy.

#### 4.3.6 Zapis danych nauczyciela do pdf

Użytkownik ma możliwość zapisu danych nauczyciela po naciśnięciu przycisku „Zapisz do pdf” [Rys 10], po naciśnięciu przycisku dane zostaną zapisane do pdf [Rys 25] oraz wyświetli się komunikat [Rys 17].

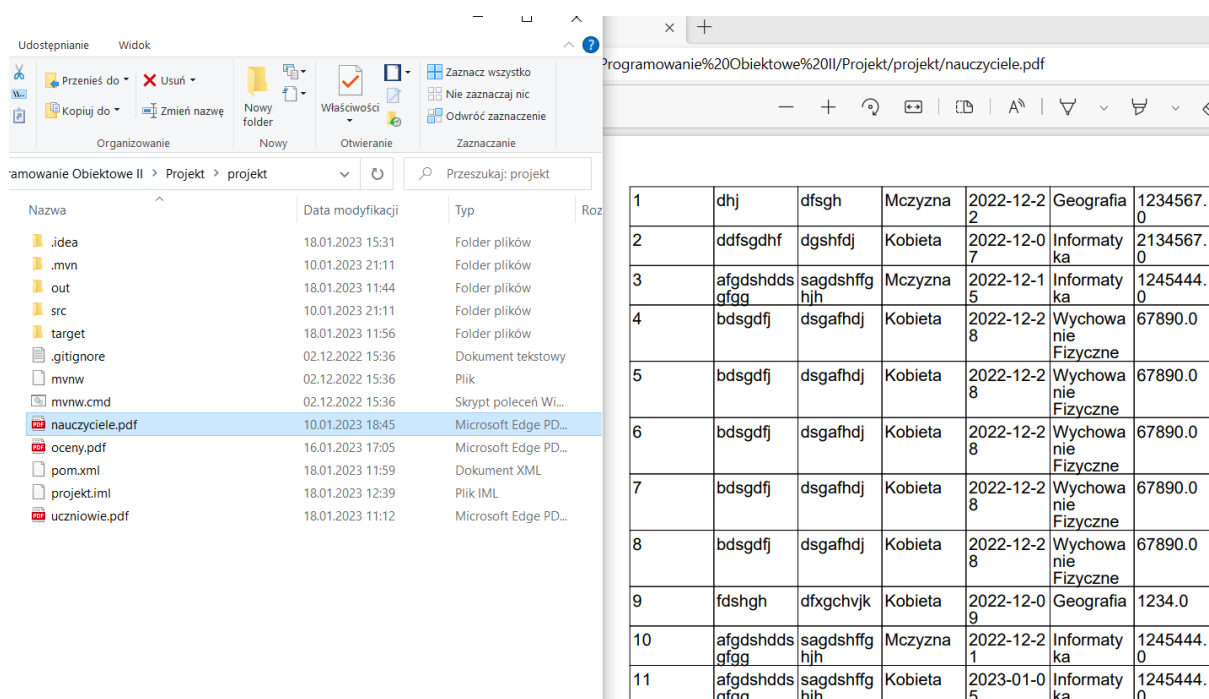


#### 4.3.7 Wyczyszczenie pul do wprowadzania danych

Użytkownik ma możliwość wyczyszczenia pul za pomocą wciśnięcia przycisku „Wyczyść” [Rys 10], po naciśnięciu przycisku nastąpi wyczyszczenie pul oraz wyświetli się komunikat [Rys 17].

#### 4.3.8 Zaktualizowanie tabeli

Użytkownik ma możliwość zaktualizowania tabeli za pomocą wciśnięcia przycisku „Zaktualizuj” [Rys 10], po naciśnięciu przycisku nastąpi zaktualizowanie tabeli.



1	dhj	dfsg	Mczyzna	2022-12-22	Geografia	1234567.0
2	ddfsgdhf	dgshfdj	Kobieta	2022-12-07	Informatyka	2134567.0
3	afgdshdds gfgg	sagdsghfj hjh	Mczyzna	2022-12-15	Informatyka	1245444.0
4	bdsgdfj	dsgafhdj	Kobieta	2022-12-28	Wychowanie Fizyczne	67890.0
5	bdsgdfj	dsgafhdj	Kobieta	2022-12-28	Wychowanie Fizyczne	67890.0
6	bdsgdfj	dsgafhdj	Kobieta	2022-12-28	Wychowanie Fizyczne	67890.0
7	bdsgdfj	dsgafhdj	Kobieta	2022-12-28	Wychowanie Fizyczne	67890.0
8	bdsgdfj	dsgafhdj	Kobieta	2022-12-28	Wychowanie Fizyczne	67890.0
9	fdshgh	dfxgchvj	Kobieta	2022-12-09	Geografia	1234.0
10	afgdshdds gfgg	sagdsghfj hjh	Mczyzna	2022-12-21	Informatyka	1245444.0
11	afgdshdds afqg	sagdsghfj hjh	Kobieta	2023-01-05	Informatyka	1245444.0

[Rys 25] Dane nauczyciela zapisane w pdf

### 4.4 Lista funkcjonalności panelu zarządzania wydarzeniami

#### 4.4.1 Przeglądanie danych w panelu wydarzenia

Panel wydarzenia umożliwia użytkownikowi przeglądanie wprowadzonych do bazy danych wydarzeń.

#### 4.4.2 Wyszukiwanie danych w panelu wydarzenia

Panel wydarzenia umożliwia użytkownikowi wyszukiwanie wydarzeń po danych wprowadzonych do bazy danych, podczas wpisywania danych poszukiwanego wydarzenia widok wydarzeń dynamicznie aktualizuje wydarzenia w tabeli na podstawie wpisanej danej [Rys 26]


The screenshot shows the 'Dziennik Szkolny' application interface. On the left is a red sidebar with a user profile icon and the text 'Witaj!'. At the bottom of the sidebar are buttons for 'Cofnij' (Back) and 'Wyloguj się' (Logout). The main content area has a search bar at the top with the text 'waka'. Below the search bar is a table with two columns: 'Wydarzenie' (Event) and 'Data' (Date). The table contains one row with the value 'wakacje' (vacations) in the 'Wydarzenie' column and '2023-07-03' in the 'Data' column. Below the table are three input fields: 'Id wydarzenia:', 'Wydarzenie:', and 'Data:'. To the right of these fields are six buttons: 'Dodaj' (Add), 'Usuń' (Delete), 'Edytuj' (Edit), 'Wyczyść' (Clear), 'Zaktualizuj' (Update), and 'Zapisz do pdf' (Save to pdf).


Wydarzenie	Data
wakacje	2023-07-03

[Rys 26] Wyszukiwanie danych w panelu wydarzenia

#### 4.4.3 Wprowadzanie danych w panelu wydarzenia

Podczas wprowadzania informacji o wydarzeniach do bazy danych wszystkie pola muszą zostać poprawnie uzupełnione w przeciwnym razie zostanie wyświetlony stosowny komunikat [Rys 27]. Natomiast po wprowadzeniu wydarzenia w poprawny sposób i następnym naciśnięciu przycisku dodaj wyświetlony zostanie stosowny komunikat [Rys 17] a wydarzenia zostanie zaktualizowana w bazie.


Dziennik Szkolny


Witaj!

Cofnij
Wyloguj się

Wydarzenie	Data
wakacje	2023-07-03

Id wydarzenia:   
Źle wprowadzone wydarzenie!

Wydarzenie:   
Źle wprowadzona data !

Data:

Dodaj
Usuń
Edytuj
Wyczyść
Zaktualizuj
Zapisz do pdf

[Rys 27] Walidacja panelu wydarzenia

#### 4.4.4 Edytowanie wydarzeń

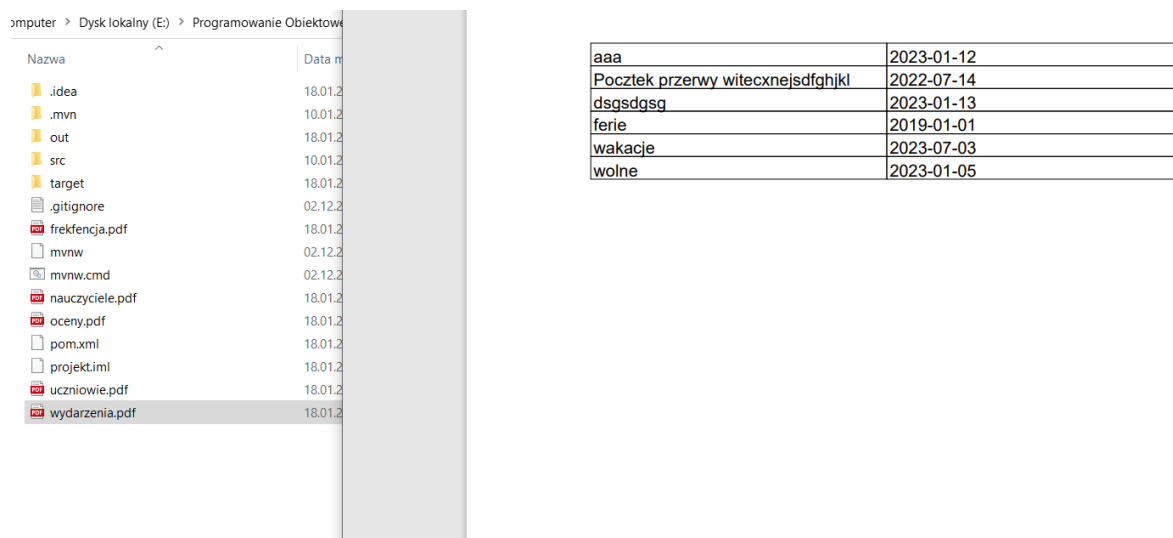
Użytkownik ma możliwość edycji danych o wydarzeniach po naciśnięciu przycisku „Edytuj” [Rys 11], po naciśnięciu przycisku jeżeli dane zostały wprowadzone prawidłowo dane zostaną zmienione oraz wyświetli się komunikat [Rys 17], natomiast jeżeli dane zostały wprowadzone w sposób nie prawidłowy zostanie wyświetlona informacja o nie prawidłowych danych [27] w zależności jaka dana została źle wprowadzona.

#### 4.4.5 Usuwanie wydarzeń

Podczas usuwania wydarzeń z bazy danych wszystkie pola muszą zostać poprawnie uzupełnione i nie mogą pozostać puste w przeciwnym razie zostanie wyświetlony stosowny komunikat Rys[27]. Natomiast po wprowadzeniu danych o wydarzeniu w poprawny sposób i następnym naciśnięciu przycisku usuń wyświetlony zostanie stosowny komunikat Rys[17] a wydarzenie zostanie usunięte z bazy.

#### 4.4.6 Zapis danych do pdf w panelu wydarzenia

Użytkownik ma możliwość zapisu wydarzeń ucznia po naciśnięciu przycisku „Zapisz do pdf” [Rys 28], po naciśnięciu przycisku dane zostaną zapisane do pdf [Rys] oraz wyświetli się komunikat [Rys 17].



[Rys 28] Zapis do pdf panelu wydarzenia

#### 4.4.7 Wyczyszczenie pul do wprowadzania danych

Użytkownik ma możliwość wyczyszczenia pul za pomocą wciśnięcia przycisku „Wyczyść” [Rys 11], po naciśnięciu przycisku nastąpi wyczyszczenie pul oraz wyświetli się komunikat [Rys 17].

#### 4.4.8 Zaktualizowanie tabeli

Użytkownik ma możliwość zaktualizowania tabeli za pomocą wciśnięcia przycisku „Zaktualizuj” [Rys 11], po naciśnięciu przycisku nastąpi zaktualizowanie tabeli oraz wyświetli się komunikat [Rys 17].

## 5. Bezpieczeństwo bazy danych

5.1 Bezpieczeństwo bazy danych jest głównie zapewnianie przez walidację występującą w samej aplikacji. Część walidacji danych została już omówiona w punkcie 4. Funkcjonalność np. weryfikowanie czy wpisywane dane w pola tekstowe są zgodne z ich założeniem dla przykładu czy hasło ma 9 znaków w tym przynajmniej 1 dużą liter i przynajmniej 1 liczbę.

Sama baza posiada zabezpieczenia pól tak aby, tabela łącząca ocena nie mogła istnieć bez ucznia oraz id\_lekcji itp.

## 6. Dokumentacja API

### 6.1 Funkcja otwierająca nowe okno aplikacji

Funkcja na Rys[29] otwiera nowe okno aplikacji tworzy nowy stage ustala jego rozmiar na stały który nie może być rozciągnięty przez użytkownika.

```
@FXML
private void handleBottomClick(javafx.event.ActionEvent mouseEvent) throws IOException {

    if (mouseEvent.getSource() == uczniowie) {
        main_form.getScene().getWindow().hide();
        LoadStages( fxml: "menuUczen.fxml");
    } else if (mouseEvent.getSource() == nauczyciele) {
        main_form.getScene().getWindow().hide();
        LoadStages( fxml: "menuNauczyciel.fxml");
    } else if (mouseEvent.getSource() == kalendarz_wydarzen) {
        main_form.getScene().getWindow().hide();
        LoadStages( fxml: "menuKalendarzWydzrzen.fxml");
    } else if (mouseEvent.getSource() == wyloguj) {
        main_form.getScene().getWindow().hide();
        LoadStages( fxml: "ekranLogowania.fxml");
    }
}

4 usages
private void LoadStages(String fxml) {
    try {
        FXMLLoader x = new FXMLLoader(getClass().getResource(fxml));
        Stage stage = new Stage();
        Parent root = x.load();
        stage.initStyle(StageStyle.UNDECORATED);
        stage.setScene(new Scene(root));
        stage.show();
    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}
```

[Rys 29] Funkcja otwierająca nowe okno

## 6.2 Funkcja dodająca nowe dane do bazy danych

Funkcja dodająca nowe dane [Rys 30] wysyła przy pomocy Hibernate dane do bazy danych gdzie są następnie zapisywane.

Najpierw zachodzi konfiguracja z klasą która wskazuje na daną tabelę w bazie danych gdzie zapisywane są dane następnie otwierana jest sesja za pomocą funkcji Session następnie rozpoczynana jest transakcja później tworzony jest nowy obiekt na podstawie którego pobierane są dane następnie następuje ustawienie danych na te które zostały wprowadzone przez użytkownika.

Później następuje wysłanie informacji o dodaniu wskazanych danych przestanie transakcji i zamknięcie sesji.

```
public void setData() {
    Configuration config = new Configuration().configure();
    config.addAnnotatedClass(Nauczyciel.class);

    StandardServiceRegistryBuilder builder =
        new StandardServiceRegistryBuilder().applySettings(config.getProperties());
    SessionFactory factory = config.buildSessionFactory(builder.build());

    Session session = factory.openSession();
    Transaction transaction = session.beginTransaction();

    Nauczyciel x = new Nauczyciel();

    if (isWypłataOk(wypłata_text_field.getText()) && isIndexOk(index_text_field.getText()) && isImieOk(imie_text_field.getText()) &&
        isNazwiskoOk(nazwisko_text_field.getText()) && data_urodzenia_date_picker.getValue() != null &&
        prowadzony_przedmiot_combobox.getSelectionModel().getSelectedIndex() != -1 &&
        plec_combobox.getSelectionModel().getSelectedIndex() != -1) {

        x.setId(Integer.parseInt(index_text_field.getText()));
        x.setImie(imie_text_field.getText());
        x.setNazwisko(nazwisko_text_field.getText());
        x.setPlec(plec_combobox.getValue().toString());
        x.setData_urodzenia(data_urodzenia_date_picker.getValue());
        x.setProwadzony_przedmiot(prowadzony_przedmiot_combobox.getValue().toString());
        x.setWypłata(Double.parseDouble(wypłata_text_field.getText()));

        session.persist(x);
        transaction.commit();
        session.close();
        LoadStages();
    }
}
```

[Rys 30] Funkcja dodająca nowe dane do bazy danych

## 6.3 Funkcja usuwająca dane w bazie danych

Funkcja usuwająca dane w bazie danych [Rys 31] wysyła przy pomocy Hibernate zapytanie do bazy danych gdzie odnajdywane są wprowadzone dane a następnie są one usuwane z bazy.

Najpierw zachodzi konfiguracja z klasą która wskazuje na daną tabelę w bazie danych gdzie zapisywane są dane następnie otwierana jest sesja za pomocą funkcji Session następnie rozpoczynana jest transakcja później tworzony jest nowy obiekt na podstawie którego pobierane są dane następnie następuje ustawienie danych na te które zostały wprowadzone przez użytkownika.

Później następuje wysłanie informacji o usunięciu wskazanych danych przesłanie transakcji i zamknięcie sesji.

```
public void deleteData() {
    Configuration config = new Configuration().configure();
    config.addAnnotatedClass(Nauczyciel.class);

    StandardServiceRegistryBuilder builder =
        new StandardServiceRegistryBuilder().applySettings(config.getProperties());
    SessionFactory factory = config.buildSessionFactory(builder.build());

    Session session = factory.openSession();
    Transaction transaction = session.beginTransaction();

    Nauczyciel x = new Nauczyciel();

    if (isWyplataOk(wyplata_text_field.getText()) && isIndexOk(index_text_field.getText()) && isImieOk(imie_text_field.getText()) &&
        isNazwiskoOk(nazwisko_text_field.getText()) && data_urodzenia_date_picker.getValue() != null &&
        prowadzony_przedmiot_combobox.getSelectionModel().getSelectedIndex() != -1 &&
        plec_combobox.getSelectionModel().getSelectedIndex() != -1) {

        x.setId(Integer.parseInt(index_text_field.getText()));
        x.setImie(imie_text_field.getText());
        x.setNazwisko(nazwisko_text_field.getText());
        x.setPlec(plec_combobox.getValue().toString());
        x.setData_urodzenia(data_urodzenia_date_picker.getValue());
        x.setProwadzony_przedmiot(prowadzony_przedmiot_combobox.getValue().toString());
        x.setWyplata(Double.parseDouble(wyplata_text_field.getText()));

        session.delete(x);

        transaction.commit();
        session.close();
        LoadStages();
    }
}
```

[Rys 31] Funkcja usuwająca dane w bazie danych

#### 6.4 Funkcja aktualizująca dane w bazie danych

Funkcja aktualizująca dane w bazie danych Rys[32] wysyła przy pomocy Hibernate'a zapytanie do bazy danych gdzie odnajdywane są wprowadzone dane a następnie są one aktualizowane na nowe.

Najpierw zachodzi konfiguracja z klasą która wskazuje na daną tabelę w bazie danych gdzie zapisywane są dane następnie otwierana jest sesja za pomocą funkcji Session następnie rozpoczynana jest transakcja później tworzony jest nowy obiekt na podstawie którego pobierane są dane następnie następuje ustawienie danych na te które zostały wprowadzone przez użytkownika.

Później następuje wysłanie informacji o zaktualizowaniu wskazanych danych przesłanie transakcji i zamknięcie sesji.

```

public void editData() {
    Configuration config = new Configuration().configure();
    config.addAnnotatedClass(Nauczyciel.class);

    StandardServiceRegistryBuilder builder =
        new StandardServiceRegistryBuilder().applySettings(config.getProperties());
    SessionFactory factory = config.buildSessionFactory(builder.build());

    Session session = factory.openSession();
    Transaction transaction = session.beginTransaction();

    Nauczyciel x = new Nauczyciel();

    if (isWypłataOK(wypłata_text_field.getText()) && isIndexOK(index_text_field.getText()) && isImieOK(imie_text_field.getText()) &&
        isNazwiskoOK(nazwisko_text_field.getText()) && data_urodzenia_date_picker.getValue() != null &&
        prowadzony_przedmiot_combobox.getSelectionModel().getSelectedIndex() != -1 &&
        plec_combobox.getSelectionModel().getSelectedIndex() != -1) {

        x.setId(Integer.parseInt(index_text_field.getText()));
        x.setImie(imie_text_field.getText());
        x.setNazwisko(nazwisko_text_field.getText());
        x.setPlec(plec_combobox.getValue().toString());
        x.setData_urodzenia(data_urodzenia_date_picker.getValue());
        x.setProwadzony_przedmiot(prowadzony_przedmiot_combobox.getValue().toString());
        x.setWypłata(Double.parseDouble(wypłata_text_field.getText()));

        session.update(x);
        transaction.commit();
        session.close();
        LoadStages();
    }
}

```

[Rys 32] Funkcja aktualizująca dane w bazie danych

## 6.5 Funkcja zaczytująca dane w bazie danych

Funkcja loadAllData [Rys 33] posiada typ dynamiczny na którym bazuje dzięki czemu możliwe jest pobieranie w funkcji fetchData danych o różnych typach.

```

public void fetchData() {
    Configuration config = new Configuration().configure();
    config.addAnnotatedClass(Nauczyciel.class);

    StandardServiceRegistryBuilder builder =
        new StandardServiceRegistryBuilder().applySettings(config.getProperties());
    SessionFactory factory = config.buildSessionFactory(builder.build());

    Session session = factory.openSession();
    Transaction transaction = session.beginTransaction();

    nauczyciel = loadAllData(Nauczyciel.class, session);

    transaction.commit();
    session.close();
}

```



```

1 usage
private static <T> List<T> loadAllData(Class<T> type, Session session) {
    CriteriaBuilder builder = session.getCriteriaBuilder();
    CriteriaQuery<T> criteria = builder.createQuery(type);
    criteria.from(type);
    List<T> data = session.createQuery(criteria).getResultList();
    return data;
}

```

[Rys 33] Funkcja zaczytująca dane w bazie danych

## 6.6 Wyszukiwanie danych

Funkcja wyszukaj [Rys 34] porównuje dane wprowadzone przez użytkownika z tymi zawartymi w bazie danych i wyszukuje ten dane odpowiadające parametrom wprowadzonym przez użytkownika.

```

public void wyszukanie() {
    FilteredList<Nauczyciel> filteredList = new FilteredList<>(observableList, e -> true);

    wyszukaj.textProperty().addListener((Observable, oldValue, newValue) -> {
        filteredList.setPredicate(predicateTeacherData -> {

            if (newValue == null || newValue.isEmpty()) {
                return true;
            }

            String searchKey = newValue.toLowerCase();
            if (String.valueOf(predicateTeacherData.getId()).toString().contains(searchKey)) {
                return true;
            } else if (predicateTeacherData.getImie().toLowerCase().contains(searchKey)) {
                return true;
            } else if (predicateTeacherData.getNazwisko().toLowerCase().contains(searchKey)) {
                return true;
            } else if (predicateTeacherData.getPlec().toLowerCase().contains(searchKey)) {
                return true;
            } else if (predicateTeacherData.getData_urodzenia().toString().contains(searchKey)) {
                return true;
            } else if (predicateTeacherData.getProwadzony_przedmiot().toLowerCase().contains(searchKey)) {
                return true;
            } else if (String.valueOf(predicateTeacherData.getWyplyta()).toString().contains(searchKey)) {
                return true;
            } else {
                return false;
            }
        });
        SortedList<Nauczyciel> sortedList = new SortedList<>(filteredList);

        sortedList.comparatorProperty().bind(tabela.comparatorProperty());
        tabela.setItems(sortedList);
    });
}

```

[Rys 34] Wyszukiwanie

## 6.7 Czyszczenie pól

Funkcja wyczyść Rys[35] ustawia dane w polach typu TextField, ComboBox i DatePicker na te początkowe czyli te przed wprowadzeniem czegośkolwiek przez Urzytkownika

```
public void wyczyść() {  
    index_text_field.setText("");  
    imie_text_field.setText("");  
    nazwisko_text_field.setText("");  
    plec_combobox.getSelectionModel().clearSelection();  
    data_urodzenia_date_picker.setValue(null);  
    prowadzony_przedmiot_combobox.getSelectionModel().clearSelection();  
    wypłata_text_field.setText("");  
}
```

[Rys 35] funkcja czyszcząca pola

## 6.8 Zapis do pdf

Funkcja Zapis do pdf [Rys 36] wywołuje połączenie z bazą danych skąd pobierane są dane ze wszystkich kolumn za pomocą napisanego polecenia ResultSet query\_set = stm.executeQuery(SELECT \* FROM nazwa\_tabeli) następnie rozpoczynany jest proces tworzenia nowego pliku pdf do którego zaczytywana jest wskazana ilość kolumn o wskazanych nazwach w bazie danych następnie proces jest kończony i tworzony jest nowy plik pdf z danymi z bazy danych.

```

@FXML
private void pdfs()
    throws Exception {
    try {

        Class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/projekt", "root", "karol0611");
        Statement stmt = con.createStatement();
        ResultSet query_set = stmt.executeQuery("SELECT * From nauczyciel");
        Document my_pdf_report = new Document();
        PdfWriter.getInstance(my_pdf_report, new FileOutputStream("nauczyciele.pdf"));
        my_pdf_report.open();
        PdfPTable my_report_table = new PdfPTable(7);
        PdfPCell table_cell;

        while (query_set.next()) {
            String index = query_set.getString("index_nauczyciela");
            table_cell = new PdfPCell(new Phrase(index));
            my_report_table.addCell(table_cell);
            String imie = query_set.getString("imie");
            table_cell = new PdfPCell(new Phrase(imie));
            my_report_table.addCell(table_cell);
            String nazwisko = query_set.getString("nazwisko");
            table_cell = new PdfPCell(new Phrase(nazwisko));
            my_report_table.addCell(table_cell);
            String plec = query_set.getString("plec");
            table_cell = new PdfPCell(new Phrase(plec));
            my_report_table.addCell(table_cell);
            String data = query_set.getString("data_urodzenia");
            table_cell = new PdfPCell(new Phrase(data));
            my_report_table.addCell(table_cell);
            String przedmiot = query_set.getString("prowadzony_przedmiot");
            table_cell = new PdfPCell(new Phrase(przedmiot));
            my_report_table.addCell(table_cell);
            String wyplata = query_set.getString("wyplata");
            table_cell = new PdfPCell(new Phrase(wyplata));
            my_report_table.addCell(table_cell);
        }
        my_pdf_report.add(my_report_table);
        my_pdf_report.close();

        query_set.close();
        stmt.close();
        con.close();
    }
}

```

[Rys 36] funkcja zapisująca do pdf

## 6.9 Regexy

Przykładowe Regexy do walidacji danych które uniemożliwiają dowolnych danych przez użytkownika. W tej aplikacji wykorzystywane są m.in. do zmuszenia użytkownika w panelu rejestracji aby w trakcie wpisywana adresu email użył raz znaku „@” oraz aby hasło zawierało minimum 8 znaków w tym przynajmniej jedną dużą literę oraz przynajmniej jedną cyfrę.

```
3 usages
private boolean isEmailOk(String s) { return s.matches( regex: "\\w{2,24}[a-z]\\d*@[1]\\w{0,24}[a-z].*"); }

3 usages
private boolean isNazwaOk(String s) { return s.matches( regex: "\\w{2,24}[a-z].*"); }

3 usages
private boolean isHasloOk(String s) { return s.matches( regex: "(?=.*[A-Za-z])(?=.*\\d)[A-Za-z\\d]{8,}$"); }
```

[Rys 37] Regexy

## 6.10 Walidacja pul

Funkcja walidacja [Rys 38] sprawdza wykorzystując regexy oraz inne przemyślane argumenty czy dane wprowadzone przez użytkownika są prawidłowo wprowadzane do bazy danych np. czy w imieniu nie ma liczb czy też czy jakie pole które musi zostać uzupełnione przez użytkownika faktycznie zostało przez niego uzupełnione

```

public void walidacja() {
    if (!isWyplataOk(wyplata_text_field.getText()) || wyplata_text_field.getText().equals("")) {
        wyplata_alert.setVisible(true);
    } else if (isWyplataOk(wyplata_text_field.getText())) {
        wyplata_alert.setVisible(false);
    }

    if (!isImieOk(imie_text_field.getText()) || imie_text_field.getText().equals("")) {
        imie_alert.setVisible(true);
    } else if (isImieOk(imie_text_field.getText())) {
        imie_alert.setVisible(false);
    }

    if (!isNazwiskoOk(nazwisko_text_field.getText()) || nazwisko_text_field.getText().equals("")) {
        nazwisko_alert.setVisible(true);
    } else if (isNazwiskoOk(nazwisko_text_field.getText())) {
        nazwisko_alert.setVisible(false);
    }

    if (!isIndexOk(index_text_field.getText()) || index_text_field.getText().equals("")) {
        index_alert.setVisible(true);
    } else if (isIndexOk(index_text_field.getText())) {
        index_alert.setVisible(false);
    }

    if (data_urodzenia_date_picker.getValue() == null) {
        data_alert.setVisible(true);
    } else if (data_urodzenia_date_picker.getValue() != null) {
        data_alert.setVisible(false);
    }

    if (prowadzony_przedmiot_combobox.getSelectionModel().getSelectedIndex() == -1) {
        przedmiot_alert.setVisible(true);
    } else if (prowadzony_przedmiot_combobox.getSelectionModel().getSelectedIndex() != -1) {
        przedmiot_alert.setVisible(false);
    }

    if (plec_combobox.getSelectionModel().getSelectedIndex() == -1) {
        plec_alert.setVisible(true);
    } else if (plec_combobox.getSelectionModel().getSelectedIndex() != -1) {
        plec_alert.setVisible(false);
    }
}
}

```

[Rys 38] Walidacja pul

#### 6.11 Funkcje Zamknij i Zminimalizuj

Funkcja zamknij [Rys39] wywołuje `System.exit(0)` która powoduje zamknięcie okna, Funkcja zminimalizuj [Rys 39] poprzez pobranie głównego panelu na którym osadzone są pozostałe elementy przy pomocy: `main_form.getScene().getWindow()` a następnie przy pomocy: `stage.setIconified` powoduje jego zminimalizowanie.

```
public void zamknij() { System.exit( status: 0); }

public void zminimalizuj(){
    Stage stage = (Stage) main_form.getScene().getWindow();
    stage.setIconified(true);
}
```

[Rys 39] Funkcje do zamykania i minimalizowania

#### 6.12 Klasa z mapowaniem

Przykład klasy służącej do nawigowania z bazą danych [Rys 40] zawiera ona pola odpowiadające nazwą kolumn w bazie danych oraz addnotacje takie jak `@Id` określające kolumnę będącą kluczem głównym w bazie danych `@GeneratedValue` Z określoną strategią oznaczającą że dana kolumna jest autoinkrementowana poza tym występuje mapowanie `@ManyToOne` wskazujące na encje z jaką występuje mapowanie oraz `@JoinColumn` określająca jej nazwę.

```

@Entity
@Table(name = "oceny")
public class Oceny {

    5 usages
    @Id
    @Column(name = "id_oceny")
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id_oceny;

    4 usages
    @Column(name = "ocena")
    private String ocena;

    4 usages
    @ManyToOne(targetEntity = Uczeń.class)
    @JoinColumn(name = "index_ucznia")
    private Uczeń index_ucznia;

    4 usages
    @ManyToOne(targetEntity = Lekcje.class)
    @JoinColumn(name = "id_lekcje")
    private Lekcje id_lekcje;

```

[Rys 40] Klasa z mapowaniem

### 6.13 Funkcje umożliwiające odczyt i zapis ocen

Funkcja `zrupFakeOceny` [Rys 41] pobiera już istniejących uczniów z tabeli `uczniowie` i następnie tworzy z nich nową listę uczniów która zaczytywana inna (edycja w ocen ucznia w jednym przedmiocie nie powoduje zmian w ocenach z innego przedmiotu) dla każdego przedmiotu z osobna

Funkcja `getData` [Rys 42] zapisuje uczniów z listy `fakeoceny` do listy która jest wyświetlana w `TableView`.

Następnie dzięki klasie `editData` [Rys 43] możliwe jest wstawianie ocen wybranemu uczniowi. Na samym początku jednak zachodzi konfiguracja z wszystkimi klasami zmapowanymi. Następnie rozpoczynana jest sesja i transakcja następnie po wybraniu ucznia z tabeli sprawdzaniu o którego ucznia z tabeli `uczeń` chodzi konkretnie oraz z jakie lekcji wstawiana będzie ocena pobierane zostają wszystkie jego dotychczasowe otrzymane oceny a następnie ograniczenie tych ocen do ocen z wybranego przez użytkownika przedmiotu. Następnie pobierane są wszystkie oceny wpisane do `TextFieldów` przez użytkownika później następuje sprawdzenie czy dana ocena była

już wprowadzona czy też uczeń otrzymuje daną ocenę pierwszy raz i w zależności od zaistniałej sytuacji następuje albo wstawienie nowej oceny albo edycja już istniejącej. Na samym końcu następuje zamknięcie transakcji i sesji.

Aby uzyskać efekt aby zaczytywać dane w sposób jaki został przewidziany koniecznym było utworzenie dodatkowej klasy w celu jej późniejszej implementacji do TabelView.

```
public void zgrupFakeOceny() {  
    for (int i = 0; i < uczniowieList.size(); i++) {  
        Integer z = uczniowieList.get(i).getId();  
        if (!idUczniow.contains(z)) {  
            idUczniow.add(z);  
            FakeOceny temp = new FakeOceny();  
            temp.setIdUcznia(uczniowieList.get(i).getId());  
            temp.setImie(uczniowieList.get(i).getImie());  
            temp.setNazwisko(uczniowieList.get(i).getNazwisko());  
            fakeocenies.add(temp);  
        }  
    }  
  
    for (int i = 0; i < observableList.size(); i++) {  
        if (observableList.get(i).getId_lekcje().getId() != idLekcji) continue;  
        double ocena = Double.parseDouble(observableList.get(i).getOcena());  
        int indexUcznia = observableList.get(i).getIndex_ucznia().getId();  
        for (int j = 0; j < fakeocenies.size(); j++) {  
            if (fakeocenies.get(j).getIdUcznia() == indexUcznia) {  
                fakeocenies.get(j).dodajOcene(ocena);  
                break;  
            }  
        }  
    }  
}
```

[Rys 41] Utworzenie nowej listy uczniów



2 usages

```
public void getData() {  
    Stage stage = (Stage) tabela.getScene().getWindow();  
    idLekcji = (Integer) stage.getUserData();  
  
    fetchData();  
    for (Oceny temp : ocenaList) {  
        observableList.add(temp);  
    }  
    zrupFakeOceny();  
    for (FakeOceny temp : fakeocenies) {  
        fakeOcenyObservableList.add(temp);  
    }  
    tabela.setItems(fakeOcenyObservableList);  
}
```

[Rys 42] odczytywanie listy w TableView

```

public void editData() {
    try {
        Configuration config = new Configuration().configure();
        config.addAnnotatedClass(Uczen.class);
        config.addAnnotatedClass(Oceny.class);
        config.addAnnotatedClass(Lekcje.class);

        StandardServiceRegistryBuilder builder =
            new StandardServiceRegistryBuilder().applySettings(config.getProperties());
        SessionFactory factory = config.buildSessionFactory(builder.build());

        Session session = factory.openSession();
        Transaction transaction = session.beginTransaction();

        //Pobieram wybranego ucznia z tabelki
        FakeOcenaUczen fakeOcenaUczen = tabela.getSelectionModel().getSelectedItem();
        if (fakeOcenaUczen != null) {
            Uczen wybranyUczen = session.find(Uczen.class, fakeOcenaUczen.getIdUcznia());
            Lekcje lekcja = session.find(Lekcje.class, idLekcji);

            //Wszystkie oceny ucznia
            List<Oceny> x = wybranyUczen.getOceny();

            //Oceny ucznia z WYBRANEGO przedmiotu
            List<Oceny> ocenyWybranegoUcznia = new ArrayList<>();
            for (int i = 0; i < x.size(); i++) {
                if (x.get(i).getId_lekcje().getId() == idLekcji) {
                    ocenyWybranegoUcznia.add(x.get(i));
                }
            }

            //Pobieram wszystkie Oceny z TextFieldow i zapisuje do tablicy
            String[] zaktualizowaneOceny = new String[6];

            zaktualizowaneOceny[0] = ocena_text_field1.getText();
            zaktualizowaneOceny[1] = ocena_text_field2.getText();
            zaktualizowaneOceny[2] = ocena_text_field3.getText();
            zaktualizowaneOceny[3] = ocena_text_field4.getText();
            zaktualizowaneOceny[4] = ocena_text_field5.getText();
            zaktualizowaneOceny[5] = ocena_text_field6.getText();
        }
    }
}

```

[Rys 42] Konfiguracja i pobranie danych

```

//Dodawanie i aktualizowanie ocen
for (int i = 0; i < zaktualizowaneOceny.length; i++) {
    //Jesli ocena nie istnieje i w polu fieldText cos wpisano to tworzy nowa
    if (i >= ocenyWybranegoUcznia.size() && !zaktualizowaneOceny[i].equals("")) {
        Oceny temp = new Oceny(zaktualizowaneOceny[i], wybranyUczen, lekcja);
        System.out.println("nowa ocena: " + temp);
        session.persist(temp);
    }

    //Jesli ocena istnieje i zostala zmieniona w fieldtext to ja aktualizuje
    if (i < ocenyWybranegoUcznia.size() && !zaktualizowaneOceny[i].equals("") && !ocenyWybranegoUcznia.get(i).getOcena().equals(zaktualizowaneOceny[i])) {
        Oceny temp = ocenyWybranegoUcznia.get(i);
        temp.setOcena(zaktualizowaneOceny[i]);
        System.out.println("Zaktualizowana ocena: " + temp);
        session.update(temp);
    }
}

transaction.commit();
session.close();
zaktualizuj();
wyczyszc();

```

[Rys 43] Dodawanie i aktualizowanie

## 7. Przydatne Informacje

Ograniczeniem aplikacji jest możliwość odczytu w panelu wstawiana ocen tylko 6 ocen dla danego ucznia. Brak wyszukiwania istniejącej takiej samej nazwy użytkownika w panelu rejestracji. Brak paginacji oraz tworzenia nagłówków podczas tworzenia pdf-ów.

Za zalety można uznać automatyczne liczenie średniej z ocen wprowadzonych przez użytkownika oraz możliwość wyeksportowania danych zawartych w tabelach do plików pdf.