

Struktura zaimplementowanego systemu bazodanowego przedstawiona została na powyższym diagramie ERD.

Poniżej znajdują się opis wszystkich tabel wchodzących w skład systemu i widocznych na powyższym diagramie ERD:

- *User* – reprezentuję użytkowników w systemie takich jak np. uczniowie, nauczyciele czy dyrektor, którzy są odróżniani dzięki relacji z tabelą *UserRole*, w przypadku ucznia posiada uzupełnione pole *ClassID* wskazujące na przynależność do danej klasy;
- *UserRole* – wskazuję role użytkownika, ma do siebie przypisany obiekt z tabeli *RolePermission*;
- *RolePermission* – zawiera zestaw pozwoleń (obiektów *Permission*) dla danej roli;
- *Permission* – pojedyncza rola;
- *Class* – reprezentuję klasę, posiada uczniów oraz wychowawcę, profil klasy wskazuje relacja z klasą *Profile*;
- *Profile* – reprezentuję profile klas;
- *Timetable* – plan zajęć dla poszczególnych przedmiotów prowadzonych przez danego nauczyciela (relacja z tabelą *ClassSubjectTeacher*), zawiera również pole odpowiadające za zastępstwo nauczyciela (jeśli takie jest wymagane);
- *ClassSubjectTeacher* – tabela asocjacyjna wiążąca ze sobą przedmiot, nauczyciela i klasę, wykorzystana w głównej mierze w planie zajęć (*Timetable*);
- *Subject* – reprezentuję przedmiot;
- *Grade* – reprezentuje ocenę dla danego ucznia z danego przedmiotu wystawioną przez danego nauczyciela;
- *GradeValue* – reprezentuje stopnie, jest w relacji z *Grade* i została od niej odseparowana ze względów na możliwość uzyskania ocen takich jak np. 5-.

W ramach systemu stworzony został również szereg funkcji, procedur, wyzwalaczy i widoków, które zostały opisane poniżej. Ponadto w jednej z procedur skorzystano z mechanizmu transakcji.

Procedury

1. Procedura nr 1: pozwala na sprawdzenie czy dana klasa ukończyła już szkołę – wykorzystana w wyzwalaczu nr 1, 4.

2. Procedura nr 2: Sprawdza czy dana klasa w danym dniu i danej godzinie ma już zajęcia – wykorzystana w wyzwalczu nr 2.
3. Procedura nr 3: wyliczenie średniej ocen klasy.
4. Procedura nr 4: wyznaczenie planu lekcji dla klasy.
5. Procedura nr 5: wyliczenie średniej ocen ucznia.
6. Procedura nr 6: wyliczenie średniej ocen ucznia dla poszczególnych przedmiotów.
7. Procedura nr 7: dodanie zastępstwa dla danych zajęć dla danego nauczyciela.

Funkcje

1. Funkcja nr 1: weryfikuje czy dany użytkownik jest studentem – wykorzystana w wyzwalczu nr 6, 7, 8, 9.
2. Funkcja nr 2: weryfikuje czy dany użytkownik jest nauczycielem i czy ma już wychowawstwo – wykorzystana w wyzwalczu nr 12, 13.
3. Funkcja nr 3: wyznacza liczbę uczniów w klasie – wykorzystana w wyzwalczu nr 14, 15.

Wyzwalacze (trigery)

1. Trigger nr 1 – before add user (uczen): weryfikuje czy klasa do której ma trafić uczeń nie zakończyła już edukacji.
2. Trigger nr 2 – before add timetable: weryfikuje czy nowe zajęcia nie zachodzą na już istniejące.
3. Trigger nr 3 – after add class: wyznacza rok ukończenia szkoły dla tej klasy.
4. Trigger nr 4 – before update user (uczen): weryfikuje czy klasa do której ma trafić uczeń nie zakończyła już edukacji.
5. Trigger nr 5 – before update class: wyznacza rok ukończenia szkoły dla tej klasy.
6. Trigger nr 6 – before add grade: weryfikuje czy user dodający ocenę na pewno nie jest uczniem.
7. Trigger nr 7 – before update grade: weryfikuje czy user dodający ocenę na pewno nie jest uczniem.
8. Trigger nr 8 – before add grade: weryfikuje czy user otrzymujący ocenę na pewno jest uczniem.
9. Trigger nr 9 – before update grade: weryfikuje czy user otrzymujący ocenę na pewno jest uczniem.

10. Trigger nr 10 – before add grade : weryfikuje czy nauczyciel wystawiający ocenę na pewno naucza ucznia otrzymującego ocenę przedmiotu z którego ocena jest wystawiana.
11. Trigger nr 11 – before update grade : weryfikuje czy nauczyciel wystawiający ocenę na pewno naucza ucznia otrzymującego ocenę przedmiotu z którego ocena jest wystawiana.
12. Trigger nr 12 – before add class: weryfikuję czy użytkownik, który ma być wychować na pewno jest nauczycielem i czy nie ma już wychowawstwa.
13. Trigger nr 13 – before update class: weryfikuję czy użytkownik, który ma być wychować na pewno jest nauczycielem i czy nie ma już wychowawstwa.
14. Trigger nr 14 – before add user (uczen): weryfikuje czy klasa do której ma trafić uczeń nie jest pełna.
15. Trigger nr 15 - before update user (uczen): weryfikuje czy klasa do której ma trafić uczeń nie jest pełna.

Widoki

1. Widok nr 1: lista ocen danego ucznia.
2. Widok nr 2: lista ocen klasy z danego przedmiotu (dziennik).
3. Widok nr 3: plan lekcji dla danego użytkownika.
4. Widok nr 4: top 10 uczniów z najlepszą w szkole średnią łączną ze swoich przedmiotów.
5. Widok nr 5: top 10 klas z najlepszą łączną średnią ocen w szkole.
6. Widok nr 6: lista uczniów klasy, którym wychodzą zagrożenia, z listą przedmiotów zagrożonych.
7. Widok nr 7: lista uczniów kwalifikujących się do stypendium/świadectwa z paskiem za średnią ocen.
8. Widok nr 8: lista klas które uczy dany nauczyciel, z przedmiotami których tam uczy.
9. Widok nr 9: lista klas z wychowawcami i rokiem szkoły.
10. Widok nr 10: lista wszystkich ocen z wartościami i nauczycielem wystawiającym ocenę.

Transakcje

1. Transakcja w procedurze nr 7 – ochrona przed dodaniem zastępstwa dla tych samych zajęć i tego samego nauczyciela w tym samym czasie.

3. Implementacja systemu – kod źródłowy, wykorzystane narzędzia

System bazodanowy zaimplementowany został w relacyjnej bazie danych MySQL w języku SQL. Przy implementacji wykorzystano system kontroli wersji Git w którym znajduje się całościowy kod źródłowy systemu wraz z jego opisem. Ponadto przy implementacji wykorzystano IDE Microsoft Visual Studio Code oraz gotowe narzędzie do zarządzania bazą danych Workbench. Link do repozytorium z implementacją znajduje się poniżej:

[Repozytorium GitHub](#)

Proces uruchomieniowy systemu został zautomatyzowany dzięki wykorzystaniu oprogramowania Docker, które jest open-source'owe i służy do konteneryzacji. Do tego celu stworzono plik *docker-compose.yml* zawierający konfigurację serwera bazodanowego i służący do jego uruchomienia na podstawie zbudowanego obrazu. Sam obraz natomiast jest tworzony dzięki plikowi *Dockerfile*, który odpowiada za stworzenie obrazu na podstawie obrazu mysql. Ponadto tworzy on zbiorcze pliki zawierające wszystkie instrukcje SQL'owe tworzące system. Są one przechowywane w katalogach *database* oraz *database/import*.

4. Część zaimplementowana do prezentacji systemu

W ramach prezentacji stworzonego systemu bazodanowego zaimplementowane aplikacje webową w język TypeScript z wykorzystaniem frameworka React. Powstałe oprogramowanie pozwala przeprowadzić przykładowe operacje na bazie danych, a także umożliwia prezentację danych wraz z uwzględnieniem zależności między nimi.

Poniżej przedstawiono zrzuty ekranu z aplikacji wraz z opisem funkcjonalności:

- Główny widok z listą top 10 uczniów, top 10 klas, uczniów zagrożonych i uczniów starających się o stypendium

Overview

Top 10 students

Student	GPA	Class
Alissa Tabin	6.00 (from 6 grades)	5c
Blanch Adrain	6.00 (from 6 grades)	6d
Singer Tobi	6.00 (from 6 grades)	7c
Katherine De Francisco	5.75 (from 12 grades)	6d
Jada McCaffrey	5.50 (from 14 grades)	6a
Tara Platon	5.50 (from 10 grades)	6d
Elissa Hector	5.50 (from 10 grades)	6b
Keeley Adger	5.50 (from 8 grades)	6d
Olis Smith	5.50 (from 8 grades)	7d
Danick Gannon	5.50 (from 10 grades)	6d

Failing students

Student	Subject	GPA
Classie Sahibing	języki	1.82
Classie Sahibing	matematyka	1.86
Classie Sahibing	informatyka	1.78
Classie Sahibing	j. hiszpański	1.73
Classie Sahibing	historia	1.79
Classie Sahibing	biologia	1.81
Classie Sahibing	wiedza	1.76
Classie Sahibing	zajęcia z psychologią	1.89
Panpa Petoviti	przyroda	1.90

Top 10 classes

Class	Class GPA	Principal
5c	3.56	Philo Collins
6a	3.51	Brigitte Tsangnor
5b	3.45	Sybil Kinywood
5d	3.46	Clarence Gulliver
7a	3.44	Tyler Moseley
7a	3.39	Alicia Hollings
6c	3.37	Colleen Harris
6d	3.27	Melissa Bradford
5a	3.24	Conita Porters
6d	3.20	Friday Shalibur

Students eligible for scholarship

Student	GPA
Blanch Adrain	6.00
Singer Tobi	6.00
Alissa Tabin	6.00
Katherine De Francisco	5.75
Keeley Adger	5.50
Elissa Hector	5.50
Jada McCaffrey	5.50
Tara Platon	5.50
Olis Smith	5.50

- Widok z listą klas:

Classes	
Class	Perceptor
5a	Gizela Pontress
5b	Clemente Gallimore
5c	Rodie Rowth
5d	Pippo Benzie
6a	Wren Costy
6b	Christabella Penlin
6c	Sylas Keywood
6d	Finlay Shellshear
7a	Antoine Rollings
7b	Tyler McIsaac
7c	Gunar Jell
7d	Catherin Henric

- Widok z listą studentów:

Students			
Name	Email	Class	
Hobe Abbado	habbado2o@ttny.cc	5b	
Leena Abbey	labbayt5@vinaora.com	5b	
Gray Aberdalg	gaberdalg9f@geocoles.jp	6c	
Clarinda Acom	caacomv@house.gov	7d	
Keeley Adger	kadgeres@illinois.edu	9d	
Blanch Adrain	badraingu@usa.gov	6d	
Miles Adshed	madshedv@networkadvertising.org	6a	
Newton Aland	nalandel@phoca.cz	9c	
Cyrlus Alecock	calecocktp@weibo.com	8c	
Kimble Allbrook	kalbrookcr@amazon.com	7a	
Ange Allen	aallen5z@bing.com	6d	
Rosalind Allatt	rallattb8@prweb.com	5a	

- Widok z listą nauczycieli:

Name
Quentin Ridett
Marlin Clavey
Christian Jilitt
Klara Gagen
Vesley Mereweather
Hugbert Muddle
Benedetto McClean
Leigha Giocannoni
Aerielia Brenton
Yasmeen Grimwad

1 2 3 4 5 ... 10 > 10 / page

- Widok pojedynczego studenta oraz możliwość dodania nowej oceny, usunięcia lub edycji istniejącej:

Add grade for j. niemiecki

Grade:

Weight:

Issuer:

Cancel OK

Modify grade

Grade: 6

Weight:

Issuer: Valaria Alden

Cancel OK

Overview

Classes

Students

Teachers

Alejoa Tobin

Personal data

Email: atobin3h@hac123.com
Address: 7828 Prairie Rose Parkway
PESEL: 52042668276

Class

9c

Overall GPA

6

Grades

Subject	GPA	Actions		
- j. niemiecki	6	+		
Grade	Weight	Date of issue	Issued by	Actions
6	3.23	26-04-2024	Valaria Alden	
6	2.04	05-04-2024	Valaria Alden	
< 1 >				
+ plastyka	6	+		
+ historia	6	+		


Timetable

Wednesday


Start time	End time	Subject	Teacher
8:00	8:45	j. niemiecki	Valaria Alden
9:00	9:45	j. niemiecki	Valaria Alden
10:00	10:45	j. niemiecki	Valaria Alden
11:00	11:45	plastyka	Clemente Gallimore
12:00	12:45	plastyka	Clemente Gallimore
13:00	13:45	j. niemiecki	Valaria Alden
14:00	14:45	historia	Christian Jilitt
15:00	15:45	plastyka	Clemente Gallimore

> Thursday

- Widok pojedynczego nauczyciela:



[Overview](#)
[Classes](#)
[Students](#)
[Teachers](#)


Valeria Alden

Personal data

Email: valden1u@vldeshare.net
Address: 7122 Shelley Way
PESEL: 28062733443

Perceptor of

None

Taught subjects

j. niemiecki

Timetable


▼ Wednesday

Start time	End time	Subject	Class
8:00	8:45	j. niemiecki	9c
9:00	9:45	j. niemiecki	9c
10:00	10:45	j. niemiecki	9c
13:00	13:45	j. niemiecki	9c


> Thursday

> Friday

- Widok pojedynczej klasy:



[Overview](#)
[Classes](#)
[Students](#)
[Teachers](#)


Class 9c

Perceptor

[Pete Dollman](#)

Class GPA

3.56

Class size

27

Students

Name	Email
Baxie Wornher	bwornher2@scoup.io
Ranra Dishman	rdishman1v@networksolutions.com
Gus Coogan	gcoogan2d@deviantart.com
Alejia Tobin	atobin3h@hao123.com
Gwendolen Escalero	gescalero3k@stgate.com
Abby Hughtland	ahughtland4z@cpanel.net
Kyle Collyns	kcollyns56@ow.ly
Donetta Tighe	dtighe5h@goo.gl

<

1

2

3

4

>

Timetable

▼ Wednesday

Start time	End time	Subject	Teacher
8:00	8:45	j. niemiecki	Valeria Alden
9:00	9:45	j. niemiecki	Valeria Alden
10:00	10:45	j. niemiecki	Valeria Alden
11:00	11:45	plastyka	Clemente Gallimore
12:00	12:45	plastyka	Clemente Gallimore
13:00	13:45	j. niemiecki	Valeria Alden
14:00	14:45	historia	Christian Jilitt
15:00	15:45	plastyka	Clemente Gallimore

> Thursday

> Friday

5. Wnioski i możliwości dalszego rozwoju

Zaimplementowany system bazodanowy do obsługi dziennika elektronicznego w szkole gwarantuje możliwość zarządzania cyklem nauczania. Stworzone tabele pozwalają przechowywać dane o uczniach, nauczycielach, klasach, planie zajęć czy ocenach. Zaimplementowane procedury i funkcje zdecydowanie mogą pomóc przy implementacji całościowego systemu pełniącego rolę dziennika elektronicznego (np. aplikacja webowa czy mobilna). Wyzwalacze, które zostały stworzone pozwolą natomiast na uniknięcie potencjalnych błędów przy wykonywaniu różnych operacji na bazie danych, np. zbyt duże klasy, błędne wystawianie ocen itp. Ponadto możliwość korzystania z zapisanych w systemie widoków zdecydowanie przyspieszy implementację, a także i samo działanie wykonywania kolejnych zapytań wymaganych w dzienniku elektronicznym.

Wybór relacyjnej bazy danych zamiast nierelacyjnej był dobrym krokiem przy jej projektowaniu. Z racji dużej ilości powiązań w projekcie dziennika elektronicznego nierelacyjny silnik bazodanowy mógłby okazać się mniej efektywny. Ponadto w przypadku samej implementacji systemu w przyszłości ten wybór może zdecydowanie przyspieszyć i ułatwić ten proces. Dokonując wyboru tego typu bazy kierowano się również możliwością tworzenia wyzwalaczy i widoków co może zmniejszyć ilość pracy przy implementacji końcowego systemu.

Zaimplementowana część do prezentacji napisana w języku TypeScript z wykorzystaniem frameworka React pomaga w zobrazowaniu działania stworzonego systemu bazodanowego. Operację, która ona oferuje (wyświetlanie, dodawanie, modyfikacje czy usuwanie danych) pomagają w lepszy sposób przedstawić jego strukturę i przykładowe użycie.

Jak już wspomniano wcześniej, stworzony system bazodanowy może zostać wykorzystany w końcowym systemie pełniącym rolę dziennika elektronicznego. Ponadto zaimplementowana część do prezentacji może być jego punktem wejściowym. Jeśli chodzi o samą bazę danych to może okazać się przydatne jego rozszerzenie o funkcjonalność sprawdzanie obecności uczniów. Do tego celu należałoby wprowadzić nową tabelę przechowującą te dane i będącą w relacji z tabelami *User* oraz *Timetable*.