

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* – reprezentują 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* – wskazują role użytkownika, ma do siebie przypisany obiekt z tabeli *RolePermission*;
- *RolePermission* – zawiera zestaw pozwoleń (obiektów *Permission*) dla danej roli;
- *Permission* – pojedyncza rola;
- *Class* – reprezentują klasę, posiada uczniów oraz wychowawcę, profil klasy wskazuje relacja z klasą *Profile*;
- *Profile* – reprezentują 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* – reprezentują 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.

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 wyzwalaczu 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.

Funkcje

1. Funkcja nr 1: weryfikuje czy dany użytkownik jest studentem – wykorzystana w wyzwalaczu nr 6, 7, 8, 9.
2. Funkcja nr 2: weryfikuje czy dany użytkownik jest nauczycielem i czy ma już wychowawstwo – wykorzystana w wyzwalaczu nr 12, 13.
3. Funkcja nr 3: wyznacza liczbę uczniów w klasie – wykorzystana w wyzwalaczu 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: weryfikuje czy użytkownik, który ma być wychować na pewno jest nauczycielem i czy nie ma już wychowawstwa.
13. Trigger nr 13 – before update class: weryfikuje 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.

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 GitHub 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 | | | Top 10 classes |
| Student | GPA | Class | |
| Juliana Tobin | 6.00 (from 6 grades) | 5c | |
| Blanch Adrain | 6.00 (from 6 grades) | 6d | |
| Gergely Tóth | 6.00 (from 6 grades) | 7c | |
| Katherine De Francisco | 6.75 (from 12 grades) | 6d | |
| Jada MacCahay | 5.50 (from 16 grades) | 6a | |
| Tara Platon | 5.50 (from 16 grades) | 5d | |
| Ekhara Hector | 5.50 (from 16 grades) | 6b | |
| Kerley Adger | 5.50 (from 8 grades) | 6d | |
| Olis Smith | 5.50 (from 8 grades) | 7d | |
| Dorack Gannon | 5.50 (from 16 grades) | 6d | |
| Failing students | | | Student eligible for scholarship |
| Student | Subject | GPA | |
| Cassie Galsberg | j. wólki | 1.82 | |
| Cassie Galsberg | matematyka | 1.66 | |
| Cassie Galsberg | informatyka | 1.78 | |
| Cassie Galsberg | j. hiszpański | 1.73 | |
| Cassie Galsberg | historia | 1.78 | |
| Cassie Galsberg | biologia | 1.81 | |
| Cassie Galsberg | religia | 1.79 | |
| Cassie Galsberg | zajęcia z wychowawcą | 1.69 | |
| Ranya Pridula | przemyśle | 1.50 | |
| Class | Class GPA | Perceptor | |
| 5c | 3.58 | Pete Gushon | |
| 6a | 3.51 | Bridgette Tranter | |
| 6b | 3.48 | Sybil Keywood | |
| 6c | 3.46 | Clemente Gallimore | |
| 6a | 3.44 | Tyler McIsaac | |
| 7a | 3.39 | Antoine Rollings | |
| 6c | 3.37 | Catherin Henric | |
| 6d | 3.37 | Nemora Bradd | |
| 6a | 3.34 | Gizela Portress | |
| 6d | 3.33 | Finlay Sheltshear | |
| Student | GPA | | |
| Blanch Adrain | 6.00 | | |
| Gergely Tóth | 6.00 | | |
| Juliana Tobin | 6.00 | | |
| Katherine De Francisco | 5.75 | | |
| Kerley Adger | 5.50 | | |
| Ekhara Hector | 5.50 | | |
| Jada MacCahay | 5.50 | | |
| Tara Platon | 5.50 | | |
| Olis Smith | 5.50 | | |

- Widok z listą klas:

| Classes | |
|--|-------------------------------------|
| Class | Perceptor |
| 5a | Gizela Portress |
| 5b | Clemente Gallimore |
| 5c | Rodie Rowth |
| 5d | Pippo Benzle |
| 6a | Wren Costy |
| 6b | Christabella Pentin |
| 6c | Sybil Keywood |
| 6d | Finlay Sheltshear |
| 7a | Antoine Rollings |
| 7b | Tyler McIsaac |
| 7c | Gunar Jell |
| 7d | Catherin Henric |
| <div> <div><</div> <div>1</div> <div>2</div> <div>></div> </div> | |

- Widok z listą studentów:

| Students | | |
|---|---------------------------------|-------|
| Name | Email | Class |
| Hobie Abbado | habbado2o@tiny.cc | 5b |
| Leena Abbay | labbayt5@vinaora.com | 9b |
| Grey Aberdalgy | gaberdalgyt@geocities.jp | 6c |
| Clarinda Acorn | cacornv@house.gov | 7d |
| Kerley Adger | kadgeres@illinois.edu | 9d |
| Blanch Adrain | badrainu@usa.gov | 6d |
| Miles Adshed | madshedy@networkadvertising.org | 6a |
| Newton Aland | nalandei@phoca.cz | 9c |
| Cyrillus Alecock | calecocktp@weibo.com | 8c |
| Kimble Allbrook | kallbrookcr@amazon.com | 7a |
| Ange Allen | aallen5z@bing.com | 6d |
| Rosalind Alltatt | rafftatt6@prweb.com | 5a |
| <div> <div><</div> <div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> <div>...</div> <div>48</div> <div>></div> </div> <div>12 / page</div> | | |

- Widok z listą nauczycieli:

| Name |
|-------------------|
| Quentin Ridett |
| Martin Clavey |
| Christian Jilkt |
| Klara Gagen |
| Vivay Moreweather |
| Hughert Muddle |
| Benedetto McClean |
| Leigha Giocannoni |
| Aerhela Brennon |
| 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:

Weight:

Issuer:

Cancel OK



OverviewClassesStudentsTeachers

Alejoa Tobin

Personal data

Email: atobin09@hso123.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 | Valeria Alden |   |
| 6 | 2.04 | 05-04-2024 | Valeria Alden |   |
| + | plastyka | 6 | + | |
| + | historia | 6 | + | |

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 Jilkt |
| 15:00 | 15:45 | plastyka | Clemente Gallimore |

- Widok pojedynczego nauczyciela:




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

Valaria Alden

Personal data

Email: valdentu@slideshare.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

[Perle Dollman](#)

Class GPA

3.56

Class size

27

Students

| Name | Email |
|------------------------------------|--|
| Bavie Wornher | bwornher2@soup.io |
| Ranna Dishman | rdishman1v@networksolutions.com |
| Gus Coogan | gcoogan2d@deviantart.com |
| Aleja Tobin | atobin3h@hao123.com |
| Gwendolen Escalero | gescalero3k@sdgate.com |
| Aky Hughland | ahughland4z@rpanel.net |
| Kyle Collyns | kcollyns56@row.ly |
| Doretta Tighe | dtighe5h@goo.gl |

<

1

2

3

4

>

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

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*.