

Dokumentacja projektu

Bazy Danych 1 - Aleksander Kluczka

1. Projekt koncepcji, założenia

1.1 Temat projektu

Stworzona baza danych realizuje strukturę uczelni. Zawiera wydziały, kierunki i przedmioty, do który przypisywani są studenci oraz prowadzący zajęcia. Celem tej bazy jest klarowny i jednoznaczny rozkład wymagań oraz obowiązków pośród wszystkich studentów oraz prowadzących. Przykładowo realizowany jest proces zapisywania studentów na przedmioty obieralne.

1.2 Funkcjonalność bazy danych i wymagania użytkownika

Projekt ten ma za zadanie realizować następujące funkcje:

- Tworzenie nowych wydziałów;
- Dodawanie nowych kierunków do konkretnych wydziałów. Na jednym wydziale nie mogą znajdować się dwa kierunki o tej samej nazwie;
- Dodawanie nowych przedmiotów do istniejących kierunków;
- Dodawanie nowych prowadzących zajęcia, którzy dzieleni są według wydziałów, do których są przypisywani;
- Dodawanie nowych studentów do poszczególnych kierunków;
- Zapisywanie studentów na przedmioty (obowiązkowe i obieralne). Przedmioty nie są przydzielane automatycznie (korzystając z informacji o konkretnym kierunku i stopniu), ponieważ z założenia studenci mogą realizować indywidualny tok studiów;
- Wypisywanie list informacji zawartych w bazie.

Sam system jest stosunkowo skomplikowany (np. baza nie wspiera studentów studiujących więcej, niż jeden kierunek w danym czasie). Dużym wyzwaniem jest zaprojektowanie skomplikowanej struktury, która zachowuje logiczny sens i przy okazji pozostaje prosta w obsłudze. Dodatkowo baza powinna wspierać aktualizowanie danych użytkowników (np. zwiększanie numeru semestru, na którym obecnie znajduje się dany student, usuwanie studentów z bazy po zakończeniu toku studiów itp.).

2. Projekt diagramów (konceptualny)

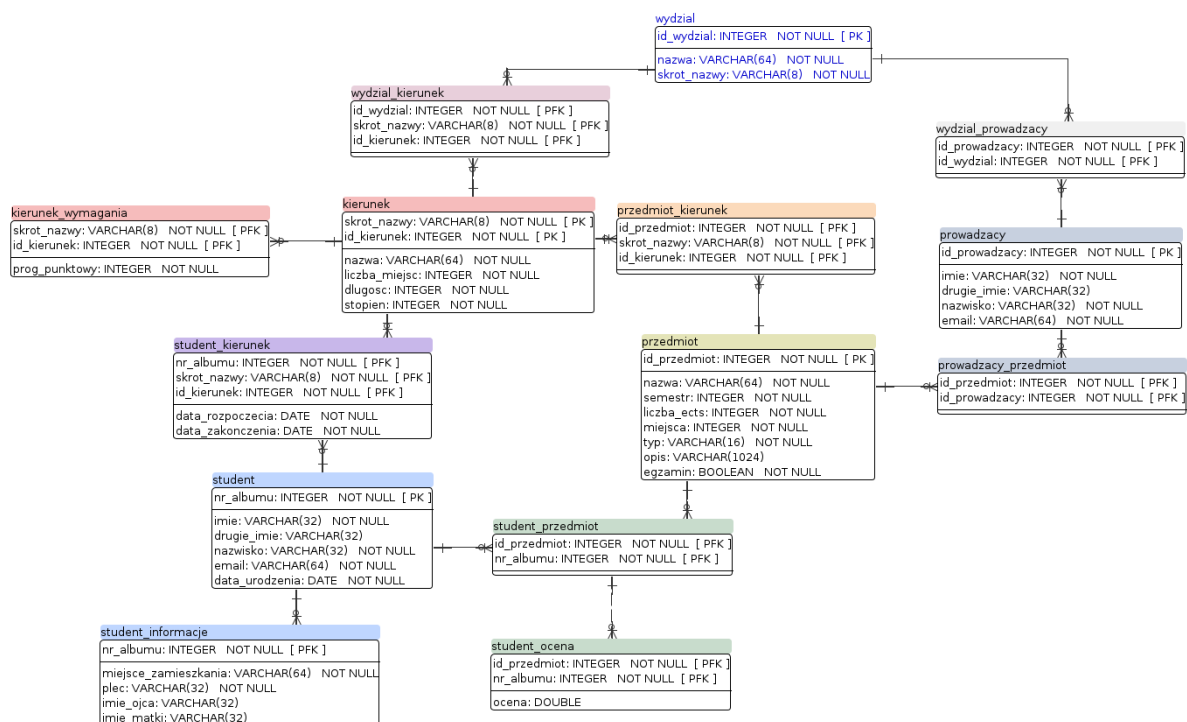
2.1 Encje i ich atrybuty

Lista encji znajdujących się w bazie danych:

- wydział: id_wydział, nazwa, skrot_nazwy
- wydział_kierunek: id_wydział, skrot_nazwy, id_kierunek

- kierunek: skrot_nazwy, id_kierunek, nazwa, liczba_miejsc, dlugosc, stopien
- kierunek_wymagania: skrot_nazwy, id_kierunek, prog_punktowy
- student_kierunek: nr_albumu, skrot_nazwy, id_kierunek, data_rozpozeczenia, data_zakonczenia
- student: nr_albumu, imie, nazwisko, drugie_imie, email, data_urodzenia
- student_informacje: nr_albumu, miejsce_zamieszkania, plec, imie_ojca, imie_matki
- student_przedmiot: id_przedmiot, nr_albumu
- student_ocena: id_przedmiot, nr_albumu, ocena
- przedmiot: id_przedmiot, nazwa, semestr, liczba_ects, miejsca, egzamin, opis, typ
- przedmiot_kierunek: id_przedmiot, skrot_nazwy, id_kierunek
- prowadzacy_przedmiot: id_przedmiot, id_prowadzacy
- prowadzacy: id_prowadzacy, imie, drugie_imie, nazwisko, email
- wydzial_prowadzacy: id_prowadzacy, id_wydzial

2.2 Relacje między encjami



3. Projekt logiczny

3.1 Tabele, klucze, indeksy

Lista tabel, kluczy oraz indeksów w bazie danych:

Student		
kolumna	typ	opis
nr_albumu	integer	klucz główny reprezentujący unikalny identyfikator studenta
imie	varchar(32)	atrybut oznaczający imię
drugie_imie	varchar(32)	atrybut oznaczający drugie imię (jeśli istnieje)
nazwisko	varchar(32)	atrybut oznaczający nazwisko
email	varchar(64)	atrybut reprezentujący adres email
data_urodzenia	date	atrybut oznaczający datę urodzenia studenta

Kierunek		
kolumna	typ	opis
skrot_nazwy	varchar(8)	klucz główny reprezentujący skróconą nazwę kierunku
id_kierunek	integer	klucz główny reprezentujący unikalny identyfikator kierunku
nazwa	varchar(64)	atrybut oznaczający pełną nazwę
liczba_miejsc	integer	atrybut reprezentujący liczbę miejsc
dlugosc	integer	atrybut oznaczający długość trwania toku studiów na danym kierunku
stopien	integer	atrybut oznaczający stopień studiów

Przedmiot		
kolumna	typ	opis
id_przedmiot	integer	klucz główny reprezentujący unikalny identyfikator przedmiotu

nazwa	varchar(64)	atrybut reprezentujący nazwę przedmiotu
semestr	integer	atrybut oznaczający numer semestru, na którym prowadzony jest przedmiot
liczba_ects	integer	atrybut reprezentujący liczbę ECTS, którą spełnia przedmiot
miejsca	integer	atrybut reprezentujący liczbę miejsc
egzamin	boolean	atrybut oznaczający konieczność zdawania egzaminu z danego przedmiotu
typ	varchar(16)	atrybut reprezentujący typ obowiązkowy lub obieralny przedmiotu
opis	varchar(1024)	atrybut oznaczający opis przedmiotu

Prowadzący		
kolumna	typ	opis
id_prowadzacy	integer	klucz główny reprezentujący unikalny identyfikator prowadzącego
imie	varchar(32)	atrybut oznaczający imię
drugie_imie	varchar(32)	atrybut oznaczający drugie imię (jeśli istnieje)
nazwisko	varchar(32)	atrybut oznaczający nazwisko
email	varchar(64)	atrybut reprezentujący adres email

Wydział		
kolumna	typ	opis
id_wydzial	integer	klucz główny reprezentujący unikalny identyfikator wydziału
nazwa	varchar(64)	atrybut oznaczające nazwę wydziału
skrot_nazwy	varchar(8)	atrybut reprezentujący skróconą nazwę wydziału

Student_kierunek		
kolumna	typ	opis

nr_albumu	integer	klucz główny (obcy) reprezentujący unikalny identyfikator studenta
skrot_nazwy	varchar(8)	klucz główny (obcy) reprezentujący skróconą nazwę kierunku
id_kierunek	integer	klucz główny (obcy) reprezentujący unikalny identyfikator kierunku
data_roz poczenia	date	atrybut oznaczający datę rozpoczęcia toku studiów
data_zakonczenia	date	atrybut oznaczający datę zakończenia toku studiów

Przedmiot_kierunek		
kolumna	typ	opis
id_przedmiot	integer	klucz główny (obcy) reprezentujący unikalny identyfikator przedmiotu
skrot_nazwy	varchar(8)	klucz główny (obcy) reprezentujący skróconą nazwę kierunku
id_kierunek	integer	klucz główny (obcy) reprezentujący unikalny identyfikator kierunku

Student_przedmiot		
kolumna	typ	opis
id_przedmiot	integer	klucz główny (obcy) reprezentujący unikalny identyfikator przedmiotu
nr_albumu	varchar(8)	klucz główny (obcy) reprezentujący unikalny identyfikator studenta

Kierunek_wymagania		
kolumna	typ	opis
skrot_nazwy	varchar(8)	klucz główny (obcy) reprezentujący skróconą nazwę kierunku
id_kierunek	integer	klucz główny (obcy) reprezentujący unikalny identyfikator kierunku
prog_punktowy	integer	atrybut oznaczający wymagany próg punktowy do dostania się na kierunek

Prowadzący_przedmiot		
kolumna	typ	opis

id_przedmiot	integer	klucz główny (obcy) reprezentujący unikalny identyfikator przedmiotu
id_prowadzacy	integer	klucz główny (obcy) reprezentujący unikalny identyfikator prowadzącego

Student_informacje		
kolumna	typ	opis
nr_albumu	integer	klucz główny (obcy) reprezentujący unikalny identyfikator studenta
miejsce_zamieszkania	varchar(64)	atrybut oznaczający miejsce zamieszkania studenta
plec	varchar(32)	atrybut oznaczający płeć studenta
imie_ojca	varchar(32)	atrybut reprezentujący imię ojca
imie_matki	varchar(64)	atrybut reprezentujący imię matki

Student_ocena		
kolumna	typ	opis
id_przedmiot	integer	klucz główny (obcy) reprezentujący unikalny identyfikator przedmiotu
nr_albumu	integer	klucz główny (obcy) reprezentujący unikalny identyfikator studenta
ocena	double	atrybut oznaczający ocenę studenta

Wydział_prowadzący		
kolumna	typ	opis
id_prowadzacy	integer	klucz główny (obcy) reprezentujący unikalny identyfikator prowadzącego
id_wydzial	integer	klucz główny (obcy) reprezentujący unikalny identyfikator wydziału

Wydział_kierunek		
kolumna	typ	opis

id_wydzial	integer	klucz główny (obcy) reprezentujący unikalny identyfikator wydziału
skrot_nazwy	integer	klucz główny (obcy) reprezentujący skróconą nazwę kierunku
id_kierunek	double	klucz główny (obcy) reprezentujący unikalny identyfikator kierunku

3.2 Widoki danych

OutKierunek: iid, nazwa, miejsca, dlugosc, stopien, wydzial, przedmioty, studenci

OutProwadzacy: id, imie, drugie imie, nazwisko, email, wydzial, przedmiot

OutPrzedmiot: id, nazwa, semestr, ect, miejsca, egzamin, typ, opis, kierunek

OutStudent: nr_albumu, imie, drugie imie, nazwisko, email, data_urodzenia, miejscowosc, plec, rozpoczecie, kierunek, stopien, wydzial

OutWydzial: id, nazwa, skrot, kierunki, prowadzacy

PrzedmiotBezProw: id_przedmiot, nazwa, semestr

ProwadzacyBezPrzed: id_prowadzacy, imie, nazwisko

LastPrzedmiot: id_przedmiot

LastProwadzacy: id_prowadzacy

LastKierunek: id_kierunek

LastWydzial: id_wydzial

LastStudent: nr_albumu

OutPrzedObieralne: id, nazwa, semestr, ect, miejsca, egzamin, typ, opis, kierunek

3.3 Analiza zbieżności funkcyjnych i normalizacja tabel

Baza danych została zaprojektowana i zaimplementowana w postaci zoptymalizowanej do 3NF.

3.4 Operacje na danych

Funkcje:

- **prj.WydzialBezKierunku** - wypisuje listę wydziałów, na których nie istnieje kierunek z podaną nazwą. Specjalnie nazwa jest brana pod uwagę, a nie ID, które przez użytkownika nie jest w ogóle widoczne. Ponadto dwa kierunki o tej samej nazwie na jednym wydziale są bardzo mylące;
- **prj.CzyKierunekNaWydziale** - sprawdza czy kierunek o danej nazwie znajduje się na konkretnym wydziale. Funkcja ta używana jest w **prj.WydzialBezKierunku**

Implementacja funkcji znajduje się w pliku *functions.sql*.

Wyzwalacze (dokładniej funkcje wywoływane przez wyzwalacze):

- **prj.walidujEmailStudenta()** - sprawdza, czy dany student z takim samym adresem email już znajduje się w bazie;
- **prj.walidujEmailProwadzacego()** - analogicznie do walidacji emaila studenta sprawdza czy istnieje już dany adres email w bazie;
- **prj.walidujWolneMiejscaNaKierunku()** - sprawdza przy dodawaniu studenta do danego kierunku czy znajdują się na nim jeszcze wolne miejsca;
- **prj.walidujWolneMiejscaNaPrzedmiocie()** - sprawdza przy dodawaniu studenta do danego przedmiotu czy znajdują się na nim jeszcze wolne miejsca;
- **prj.walidujNowyKierunek()** - sprawdza poprawność wszystkich informacji podawanych przy dodawaniu nowego kierunku;
- **prj.walidujNowyKierunek_Wymagania()** - sprawdza poprawność wszystkich informacji podawanych przy dodawaniu nowych wymagań do kierunku;
- **prj.walidujNowyWydzial()** - sprawdza poprawność wszystkich informacji podawanych przy dodawaniu nowego wydziału;
- **prj.walidujNowegoProwadzacego()** - sprawdza poprawność wszystkich informacji podawanych przy dodawaniu nowego prowadzącego;
- **prj.walidujNowyWydzial_Prowadzacy()** - sprawdza poprawność wszystkich informacji podawanych przy dodawaniu prowadzącego do wydziału;
- **prj.walidujNowegoProwadzacego_Przedmiot()** - sprawdza poprawność wszystkich informacji podawanych przy przypisywaniu prowadzącego do przedmiotu;
- **prj.walidujNowyPrzedmiot()** - sprawdza poprawność wszystkich informacji podawanych przy dodawaniu nowego przedmiotu;
- **prj.walidujNowyPrzedmiot_Kierunek()** - sprawdza poprawność wszystkich informacji podawanych przy przypisywaniu przedmiotu do istniejącego kierunku;
- **prj.walidujNowyStudentPrzedmiot()** - sprawdza poprawność wszystkich informacji podawanych przy przypisywaniu studenta do istniejącego przedmiotu;
- **prj.walidujNowaStudent_Ocena()** - sprawdza poprawność wszystkich informacji podawanych przy dodawaniu oceny dla danego studenta na danym przedmiocie;
- **prj.walidujNowegoStudenta()** - sprawdza poprawność wszystkich informacji podawanych przy dodawaniu nowego studenta;
- **prj.walidujNoweStudent_Informacje()** - sprawdza poprawność wszystkich informacji uzupełniających dodawanych do studenta;
- **prj.walidujNowyStudent_Kierunek()** - sprawdza poprawność wszystkich danych podawanych przy zapisywaniu studenta na kierunek;

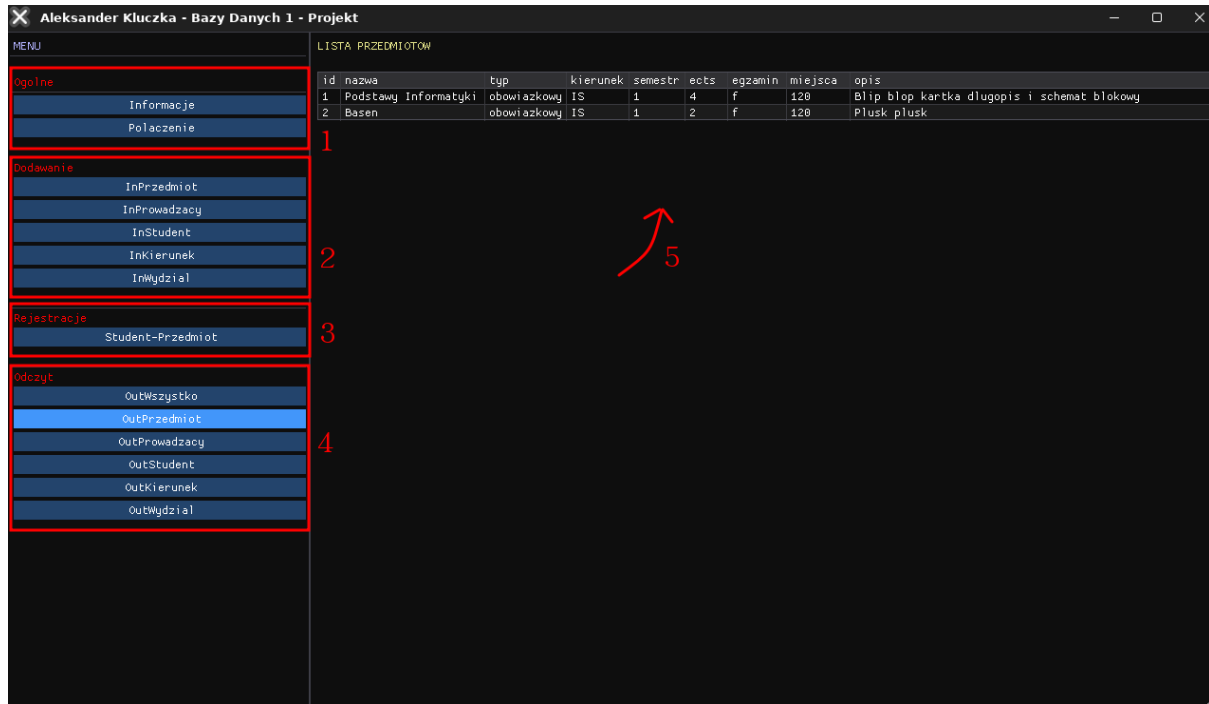
Implementacja wyzwalaczy i wykonywanych przez nich funkcji znajduje się w pliku *triggers.sql*.

4. Projekt funkcjonalny

4.1 Interfejs użytkownika - wprowadzanie i wizualizacja danych

Aplikacja graficzna zawiera predefiniowane formularze do wprowadzania danych oraz widoki do ich wyświetlania. Do jej stworzenia użyto C++ oraz bibliotek ImGui, SFML, SFML-ImGui. Za obsługę baz danych odpowiada biblioteka libpqxx.

Przykładowy wygląd programu ukazany jest na poniższym obrazie.



Oznaczenia:

1. Ogólna sekcja menu - tutaj znajdują się przyciski do uzyskania podstawowych informacji o aplikacji i sprawdzania połączenia.
2. Sekcja wprowadzania danych - w tych oknach korzysta się z formularzy do wprowadzania nowych danych. Oczywiście użytkownik jest ograniczony do wprowadzania tylko do istniejących tabel - nie ma możliwości tworzenia nowych relacji i encji.
3. Sekcja operowania na danych - tutaj wprowadzane są dane, które ściśle powiązane są z istniejącymi już danymi (dlatego stworzenie kilku pozycji w sekcji drugiej jest konieczne do poprawnego działania).
4. Sekcja wyświetlania danych - w tym miejscu znajdują się tabele, które wyświetlają wprowadzone dane. Gdyby zaistniała sytuacja, że użytkownik wprowadził nowe pozycje w sekcji drugiej i od razu chciał je wyświetlić w tej sekcji, istnieje niewielka szansa, że nie zostaną one pokazane w tabeli przed restartem aplikacji. Wynika to z implementacji funkcjonalności połączenia interfejsu z wybraną bazą danych z pomocą biblioteki libpqxx.
5. Zawartość danego okna - w zależności od wybranej pozycji w menu, znajdują się tutaj formularze do wprowadzania danych lub tabele do ich wyświetlania.

Na poniższych zdjęciach pokazane są niektóre formularze:

NOWY PRZEDMIOT

Nazwa	varchar(64)
Numer semestru	int
Liczba punktów ECTS	int
Liczba miejsc	int
<input type="checkbox"/> Egzamin	
Typ przedmiotu (obowiązkowy/obieralny)	varchar(16)
Opis	varchar(1024)

Wybierz kierunek: ▼

Po wpisaniu odpowiednich pól pojawia się przycisk wysłania nowych informacji. Nie jest on dostępny od razu, aby zapobiec niepotrzebnemu wysyłaniu kwerend do bazy danych.

NOWY STUDENT	
Numer albumu	int
Imie	varchar(32)
Drugie imie	varchar(32)
Nazwisko	varchar(32)
Email	varchar(64)
1 - + 1 - + 2000 - +	date(dd-mm-yyyy)
Miejsce zamieszkania	varchar(64)
Plec	varchar(16)
Imie Ojca	varchar(32)
Imie Matki	varchar(32)
Stopien studiow	int

NOWY KIERUNEK	
Nazwa	varchar(64)
Skrot nazwy	varchar(8)
Liczba miejsc	int
Dlugosc toku studiow (w semestrach)	int
Stopien	int
Prog punktowy	int

NOWY WYDZIAL	
Nazwa	varchar(64)
Skrot nazwy	varchar(8)

5. Dokumentacja

5.1 Wprowadzanie danych

Poza predefiniowanymi danymi wprowadzonymi na początek do bazy danych, wszystkie nowe informacje wprowadzane są ręcznie za pomocą formularzy zawartych w aplikacji graficznej.

W niektórych przypadkach formularz pozwala na wybranie opcji z dostępnej listy, aby użytkownik nie wprowadzał tych informacji manualnie.

Następujące tabele wymagają wpisywania ręcznego:

- wydział;
- kierunek;

- prowadzący;
- przedmiot;
- student.

Jeśli w formularzu zostały uzupełnione poszczególne pola (poprzez np. wybranie z listy rozwijanej), dane są wprowadzane automatycznie w tabelach:

- wydział_kierunek;
- kierunek_wymagania;
- przedmiot_kierunek;
- wydział_prowadzacy;
- student_kierunek;
- prowadzący_przedmiot;
- student_informacje;
- student_przedmiot.

5.2 Dokumentacja użytkownika - instrukcja uruchomienia

5.2.1 Gotowy plik wykonywalny

Dla systemu Windows skompilowano aplikację do gotowego do uruchomienia pliku wykonywalnego .exe. Znajduje się on w archiwum. Dodatkowo jest on dostępny do ściągnięcia pod tym linkiem: https://github.com/vis4rd/db_project-gui_2021/releases.

5.2.2 Własnoręczna kompilacja

Do kompilacji skonfigurowany został system z wykorzystaniem narzędzia CMake. Najprostszym sposobem jest ściągnięcie repozytorium z portalu GitHub za pomocą komendy:

```
git clone --recursive https://github.com/vis4rd/db_project-gui_2021.git
```

W folderze głównym repozytorium należy stworzyć folder `build/`. W folderze tym należy wywołać narzędzie CMake: `cmake -DCMAKE_BUILD_TYPE=Release ..`, a następnie generator kompilujący, który został wybrany przez CMake. Na Windows jest to zazwyczaj solucja do MS Visual Studio, natomiast na systemach Unix jest to narzędzie `make`. Konkretny generator można wybrać (np. MinGW `make`) dodając do komendy flagę: `-G "MinGW Makefiles"`.

Aplikacja korzysta z zewnętrznych bibliotek do C++ wypisanych w punkcie 5.4, dlatego wszystkie wymagania co do jej uruchomienia wynikają z ich potrzeb. Istnieje niezerowe prawdopodobieństwo, że dana maszyna z systemem Windows spełnia wszystkie wymagania bez dodatkowego zaangażowania użytkownika. Z kolei na systemach Unix niemalże na 100% trzeba będzie własnoręcznie przygotować środowisko, żeby poprawnie skompilować aplikację.

Warto wspomnieć, że biblioteki nie wymagają wcześniejszej osobnej kompilacji. Wszystkim automatycznie zajmuje się przygotowany plik CMakeLists.txt zawarty w kodzie źródłowym programu.

5.3 Dokumentacja techniczna

Dokumentacja techniczna aplikacji klienta została umieszczona w kodzie. Dodatkowo została wygenerowana wersja HTML z pomocą narzędzie Doxygen - znajduje się w folderze "doc".

5.4 Wykaz literatury

1. Strona profesora A. Dydejczyka - <https://newton.fis.agh.edu.pl/~antek/index.php?sub=dbase>.
2. Dokumentacja języka C++ - <https://en.cppreference.com/w/>.
3. Dokumentacja PostgreSQL - <https://www.postgresql.org/docs/14/index.html>.
4. Dokumentacja biblioteki libpqxx - <https://libpqxx.readthedocs.io/en/7.6.0/>.
5. Dokumentacja biblioteki ImGui - <https://imgui-test.readthedocs.io/en/latest/>.
6. Dokumentacja biblioteki SFML - <https://www.sfm1-dev.org/documentation/2.5.1/>.
7. Dokumentacja narzędzia CMake: <https://cmake.org/cmake/help/v3.22/>.
8. Forum <https://stackoverflow.com>.
9. Repozytoria GitHub:
 - a. [libpqxx](#)
 - b. [SFML](#)
 - c. [ImGui](#)
 - d. [SFML-ImGui](#)