

Dokumentacja Projektu

Biblioteka

Bazy Danych 1

Przemysław Rodzik

2022

1. Koncepcja projektu

Temat projektu

Projekt ułatwia organizacji bazy danych biblioteki, składającej się z książek, które są niepowtarzalne. Po zalogowaniu się jako klient albo pracownik ukazuje się panel z udostępnionymi funkcjonalnościami. Klient ma możliwość przeglądania książek, autorów, sprawdzania dzieł danego autora. Pracownik ma możliwość dokonywania zmian we wszystkich tabelach w bazie danych. Baza danych jest hostowana na portalu ElephantSQL.

Analiza wymagań użytkownika

Jako Użytkownik:

- Możliwość przeglądania własnych informacji konta
- Możliwość przeglądania własnych rezerwacji i dokonywanie nowych.
- Przeglądanie wszystkich książek
- Przeglądanie książek i następnie sprawdzenie wszystkich dzieł wybranego autora

Jako Pracownik:

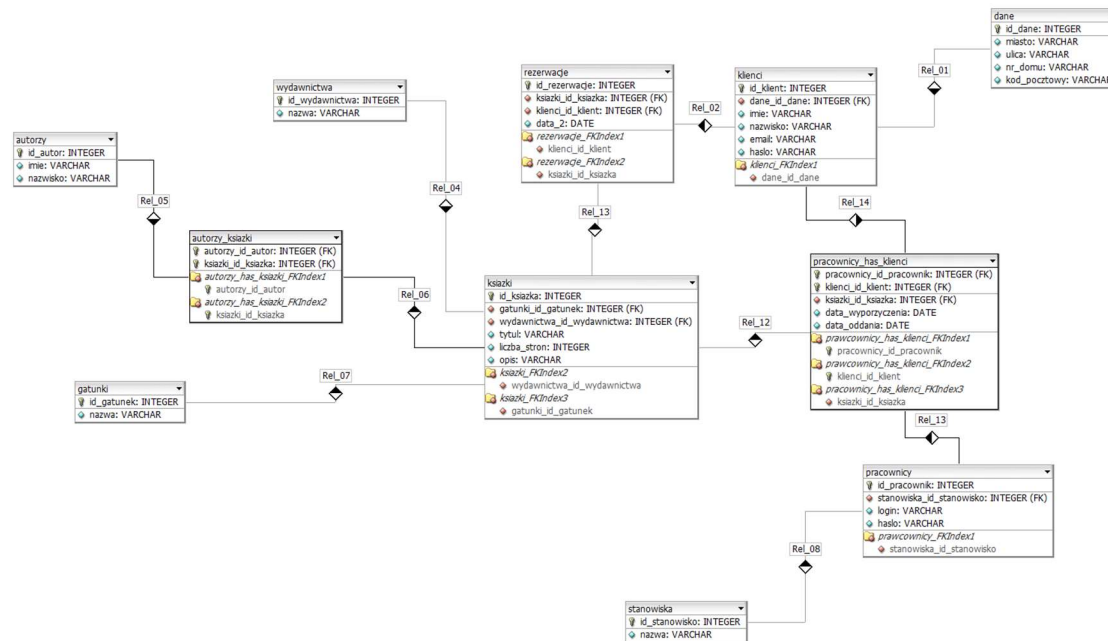
- Jest umożliwione usuwanie wszystkich informacji z tabel i dodawanie nowych
- Możliwość dodawania terminów, w których została dodana dana książka przez klienta
- Możliwość przeglądania swoich informacji i informacji wszystkich klientów i innych pracowników

Zaprojektowane funkcje:

- Wprowadzanie danych do tabeli
- Dodawanie rezerwacji
- Dodawanie terminów wydania i oddania książek
- Zestaw triggerów oraz funkcji walidujących wprowadzane dane (dane są również walidowane od strony GUI więc są dwa poziomy walidacji)
- Funkcje umożliwiające zmianę hasła przy podaniu aktualnego hasła
- Widoki ułatwiające wizualizację danych w tabelach

2. Projekt diagramów (konceptualny)

Budowa i analiza diagramu przepływu danych (ERD)



Książki:

Przechowuje informacje o książkach znajdujących się w bibliotece. Id_książka jest kluczem tabeli indywidualny dla każdej książki.

Autorzy Książki:

Przechowuje informacje o autorach jest to relacja m do n, ponieważ autor może mieć kilka książek i książka może mieć wielu autorów.

Klienci:

Są powiązani z tabelą przechowującą ich dane, ponieważ np. gdy różni członkowie rodziny mają oddzielne konta to wtedy mają ten sam adres.

Rezerwacje:

Przechuje jakich rezerwacji dokonywali klienci na podane książki.

Pracownicy_has_klienci:

Tabela przechowująca informacje o tym jakie książki zostały wydane jest relacją m do n. Rozróżniane są poprzez id_klienta, id_pracownika, który wydawał książkę oraz id_książki.

3. Projekt logiczny

Projektowanie tabel, kluczy, indeksów

```
CREATE TABLE autorzy
(
    id_autor SERIAL PRIMARY KEY,
    imie VARCHAR NULL,
    nazwisko VARCHAR NULL
);

CREATE TABLE autorzy_książki
(
    autorzy_id_autor INTEGER NOT NULL,
    książki_id_książka INTEGER NOT NULL,
    PRIMARY KEY (autorzy_id_autor, książki_id_książka)
);

CREATE TABLE gatunki
(
    id_gatunek SERIAL PRIMARY KEY,
    nazwa VARCHAR NULL
);

CREATE TABLE wydawnictwa
(
    id_wydawnictwa SERIAL PRIMARY KEY,
    nazwa VARCHAR NULL
);

CREATE TABLE książki
(
    id_książka SERIAL PRIMARY KEY,
    gatunki_id_gatunek INTEGER NOT NULL,
    wydawnictwa_id_wydawnictwa INTEGER NOT NULL,
    tytuł VARCHAR NULL,
    liczba_stron INTEGER NULL,
    opis VARCHAR NULL
);

CREATE TABLE rezerwacje
(
    id_rezerwacje SERIAL PRIMARY KEY,
    książki_id_książka INTEGER NOT NULL,
    klienci_id_klient INTEGER NOT NULL,
    data_2 DATE NULL
);

CREATE TABLE klienci
(
    id_klient SERIAL PRIMARY KEY,
    dane_id_dane INTEGER NOT NULL,
    imie VARCHAR NULL,
    nazwisko VARCHAR NULL,
    email VARCHAR NULL,
    haslo VARCHAR NULL
);

CREATE TABLE stanowiska
(
    id_stanowisko SERIAL PRIMARY KEY,
    nazwa VARCHAR NULL
);

CREATE TABLE pracownicy
(
    id_pracownik SERIAL PRIMARY KEY,
    stanowiska_id_stanowisko INTEGER NOT NULL,
    login VARCHAR NULL,
    haslo VARCHAR NULL
);
```

```

CREATE TABLE pracownicy_has_klienci
(
    pracownicy_id_pracownik INTEGER NOT NULL,
    klienci_id_klient        INTEGER NOT NULL,
    ksiazki_id_ksiazka      INTEGER NOT NULL,
    data_wyporzczenia       DATE NULL,
    data_oddania             DATE NULL,
    PRIMARY KEY (pracownicy_id_pracownik, klienci_id_klient)
);

CREATE TABLE dane
(
    id_dane          SERIAL PRIMARY KEY,
    miasto           VARCHAR NULL,
    ulica            VARCHAR NULL,
    nr_domu          VARCHAR NULL,
    kod_pocztowy     VARCHAR NULL
);

```

Słownik danych

Nazwa	Typ	Opis
Id	Key	Unikalny, niepowtarzalny numer rozróżniający wiersze w tabelach.
Autorzy_id autor	Integer	Numer rozpoznawający autora danej książki.
Ksiazki_id_ksiazka	Integer	Numer rozpoznawający książkę, aby przypisać do nich autorów.
Garunki_id_gatunek	Integer	Numer przypisujący gatunek do danej książki.
Wydawnictwa_id_wydawnictwo	Integer	Numer przypisujący wydawnictwo do danej książki.
Klienci_id_klienci	Integer	Numer przypisujący do danego wiersza odpowiadającemu mu klienta.
Stanowisko_id_stanowisko	Integer	Numer przypisujący do stanowisko do danego wiersza w tabeli.
Pracownicy_id_pracownik	Integer	Numer przypisujący pracownika do danego wiersza w tabeli.
Data_wyporzczenia	Date	Pole przechowujące datę wypożyczenia danemu klientowi.
Data_oddania	Date	Pole przechowujące datę oddania książki przez klienta. Przechowuje null, gdy książka nie została oddania.

Analiza zależności funkcyjnych i normalizacja tabel

Zoptymalizowanie tabel do 3NF zostało zapewnione na etapie projektowania bazy. Spełnione są warunki 1., 2. oraz 3. postaci normalnej, mówiących o atomowości danych, o tym, że każda tabela opisuje jeden obiekt, ma swój klucz główny, każdy niekluczowy atrybut tabeli jest zależny funkcyjnie od całego klucza głównego oraz żaden atrybut niekluczowy nie jest zależny funkcyjnie od innych atrybutów niekluczowych.

Zaprojektowanie operacji na danych

Wszystkie funkcje i trigerry pozwalające przeprowadzać walidację na wprowadzanych danych oraz funkcje pozwalające edytować dane w tabelach są umieszczone w SQL.sql. Wraz z dokumentacją kodu psql,

Również ten plik zawiera sposób w jaki zostały stworzone wszystkie tabele oraz podstawowe dane pozwalające na pracę w aplikacji.

4. Projekt funkcjonalności

-Panel logowania

Przykładowe dane logowania:

Klient:

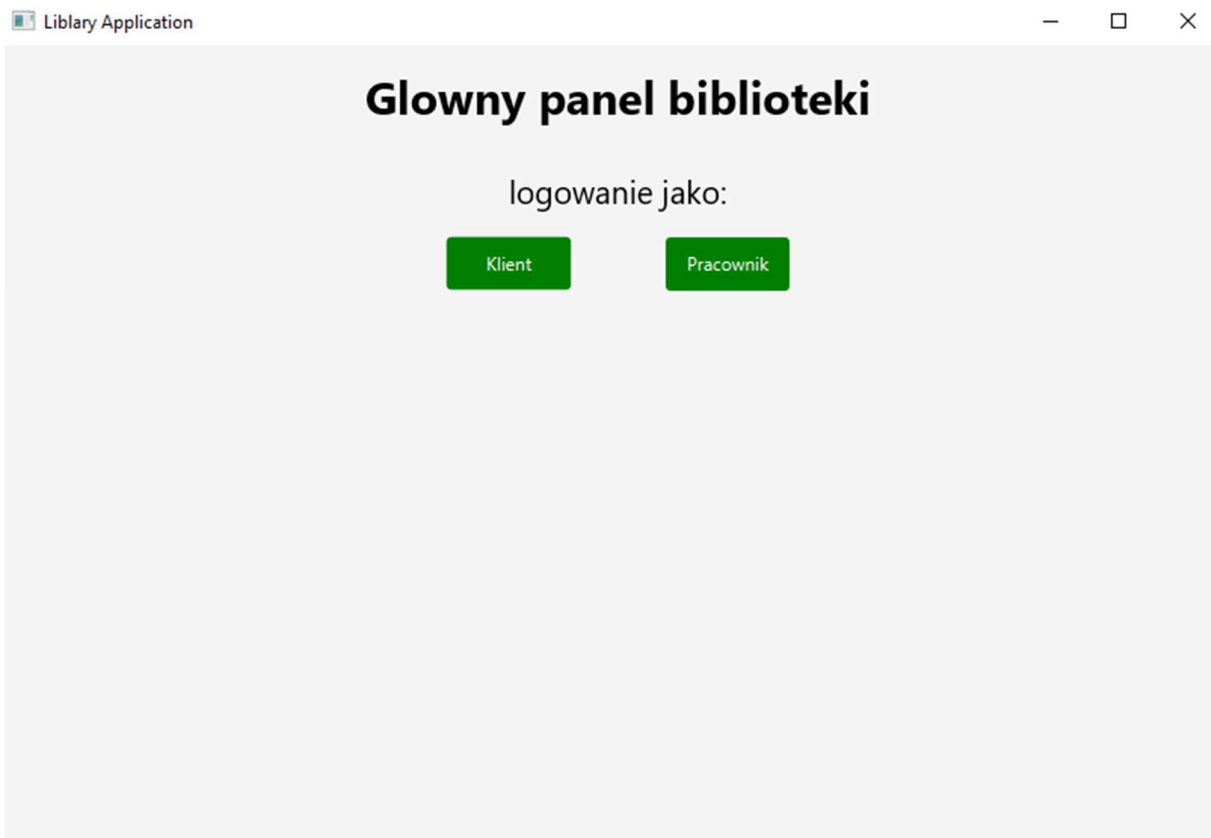
Login: test

Hasło: test

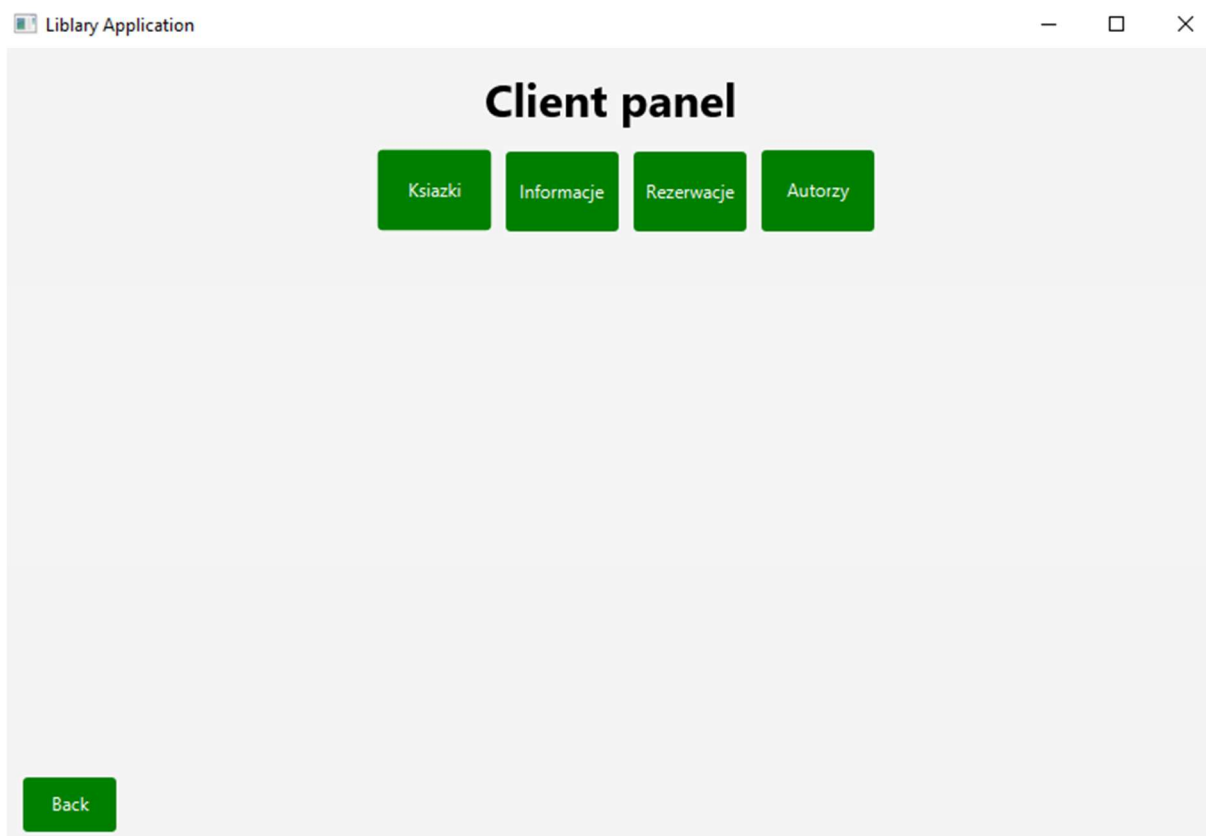
Pracownik:

Login: admin

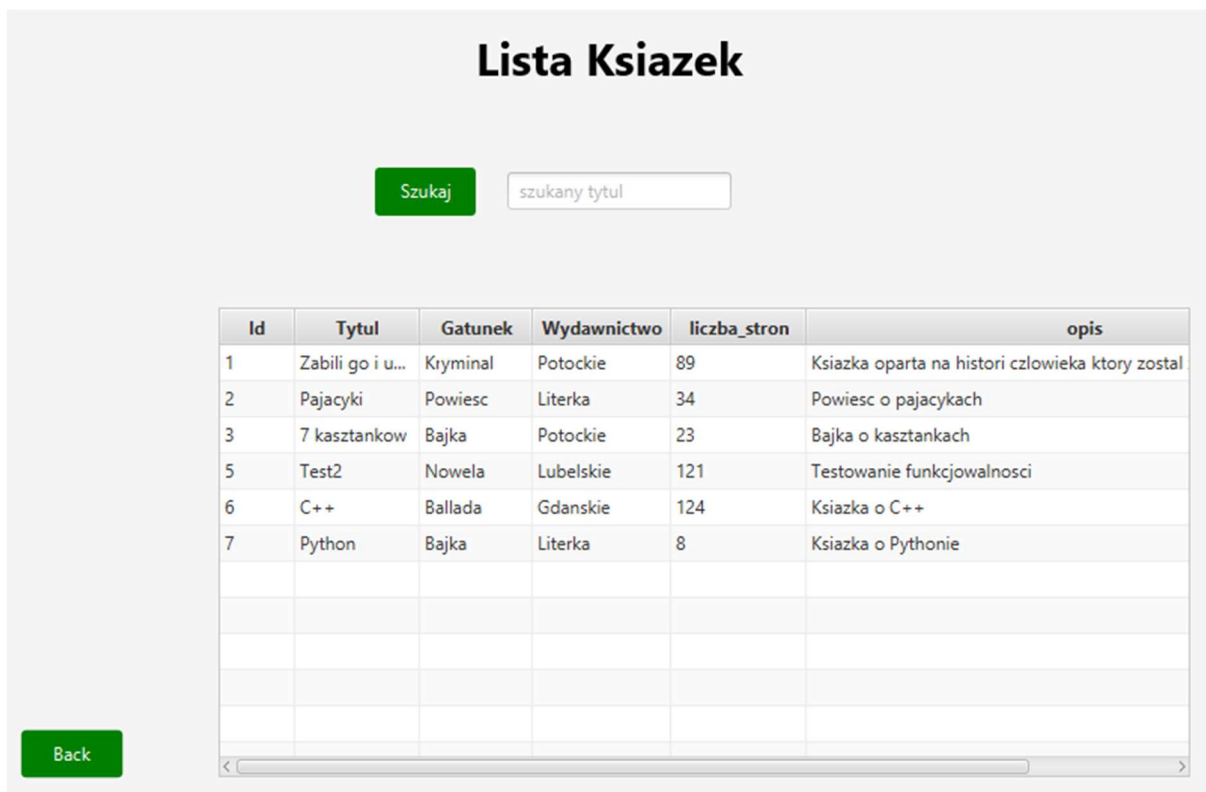
Hasło: admin



- Panel Klienta



- Przykładowy obraz interfejsu dla Klienta (Klient może przeglądać wszystkie książki i je filtrować)



- Klient ma możliwość dokonywania własnych rezerwacji na książki

Library Application

Lista Rezerwacji

Dodaj Rezerwacje

Książka

Data

Książka	Imię	Nazwisko	Data
Pajacyki	test	test	2000-01-01
Pajacyki	test	test	2001-01-01
Pajacyki	test	test	2002-02-02
Zabili go i uciakl	test	test	2003-03-03
C++	test	test	2008-04-04

Back

- Jaki i przeglądania informacji o własnym koncie

Library Application

Informacje:

Email:
test

Imię:
test

Nazwisko:
test

Miasto:
Krakow

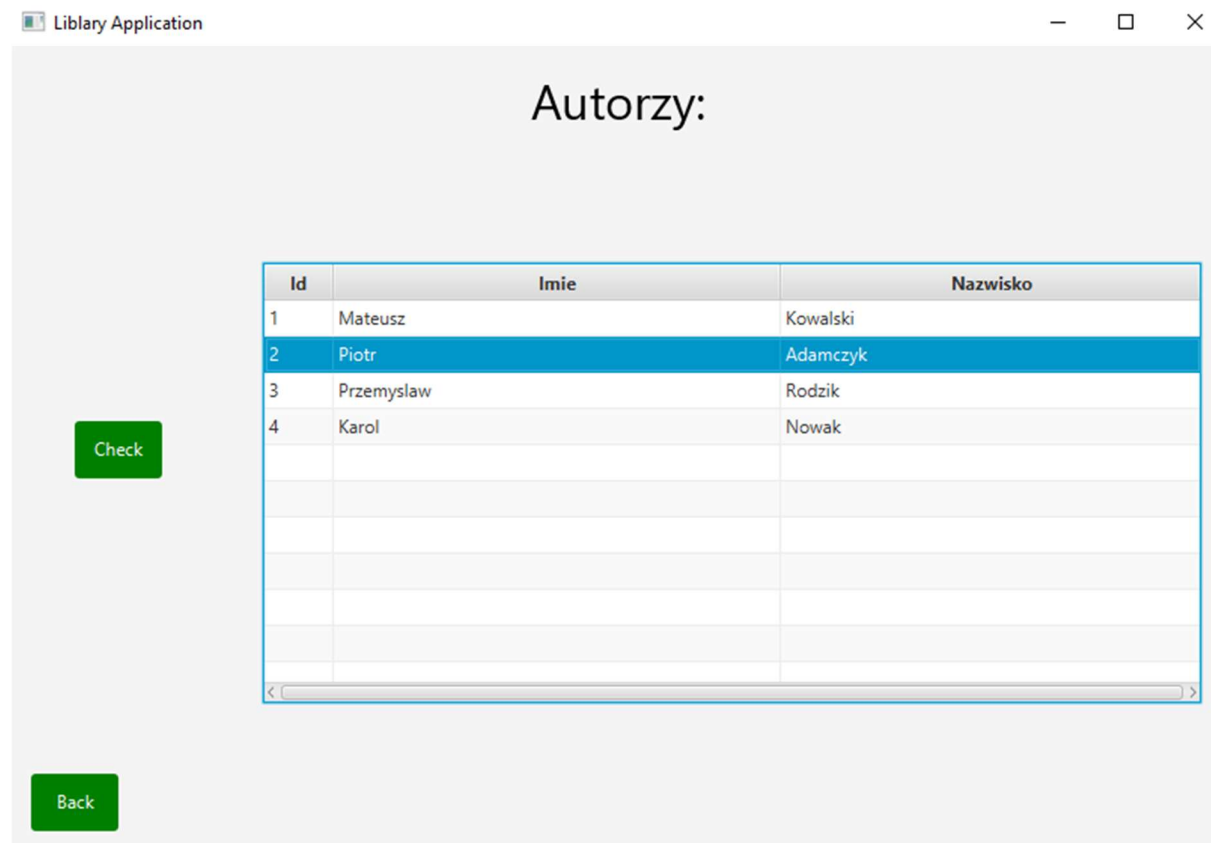
Ulica:
Czerwona

Nr domu:
34

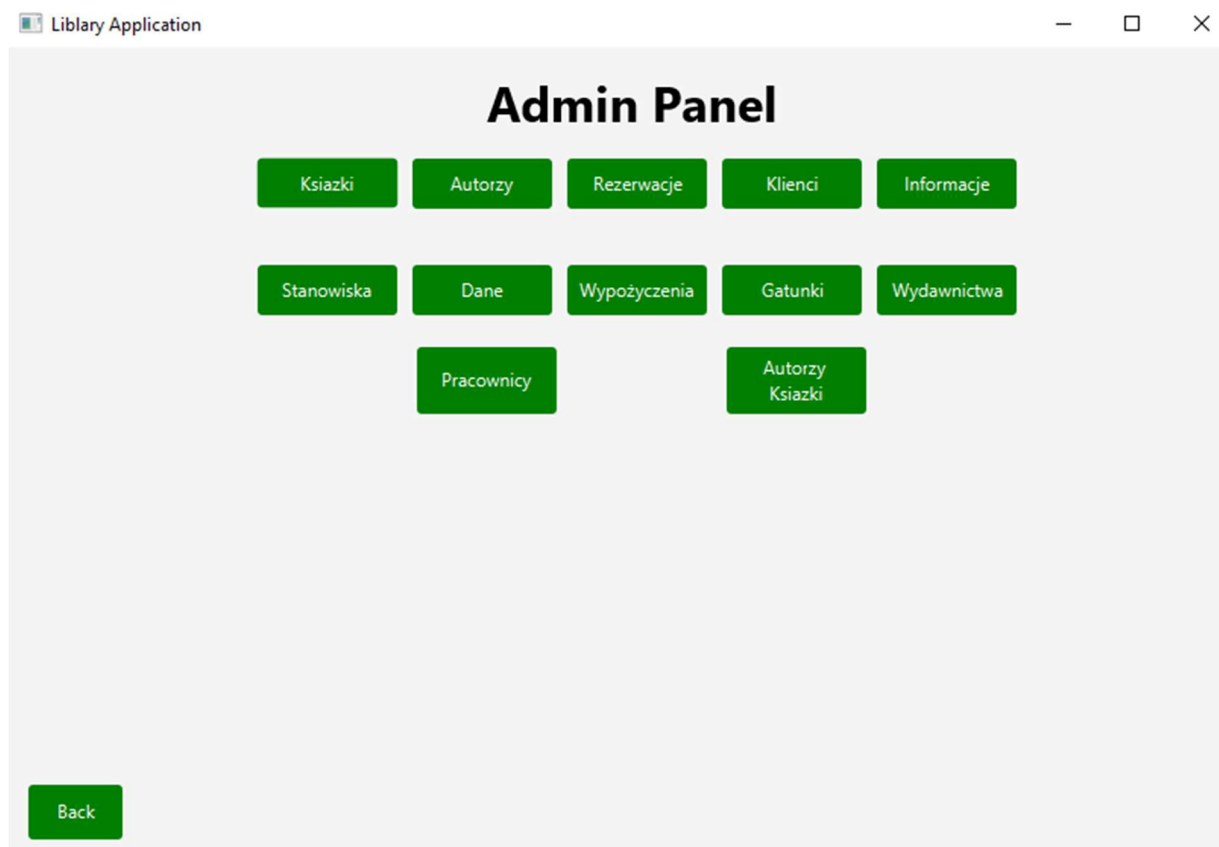
Kod_pocztowy:
30-124

Back

- Przy zaznaczonym widoku jednego autora i wciśnięciu przycisku „Check” zostaną wyświetlone wszystkie dzieła danego autora



- Pracownicy biblioteki mają znacznie zwiększoną możliwość edycji bazy



- Każdy interfejs wygląda analogicznie w zależności od ilości i typu danych w wybranej tabeli

(Usunięcie Książki skutkuje usunięciem wszelkiego śladu po danej książce rezerwacji, relacji z autorami jak i informacji kto wypożyczał daną książkę kiedykolwiek, lecz nie usunie danego autora książki, ponieważ jest możliwość, że jest autorem innej książki)

Library Application

Lista Ksiazek

Create

Id_gatunek

Id_wydawnictwo

tytul

liczba_stron

opis

Id	Id_gatunek	Id_wydawnictwo	tytul	liczba_stron	opis
1	1	1	Zabili go i uciakl	89	Ksiazka oparta na histori czlowieka ktory zos...
2	3	2	Pajacyki	34	Powiesc o pajacykach
3	2	1	7 kasztankow	23	Bajka o kasztankach
4	7	4	test	66	Test
5	4	5	Test2	121	Testowanie funkcjonalnosci
6	5	3	C++	124	Ksiazka o C++
7	2	2	Python	8	Ksiazka o Pythonie

Delete

Back

- Pracownicy mają również możliwość uaktualniania bazy wypożyczeni o datę oddania książki

Library Application

Lista Wypozycczen

Create

Update

id_pracownik

id_klient

id_ksiazka

data_wypozycczenia

data_oddania

id_pracownik	id_klient	id_ksiazka	data_wypozycczenia	data_oddania
1	1	1	2021-12-21	2021-12-31
1	2	1	2021-12-26	2022-01-15
3	5	5	2001-04-04	2002-04-04
2	1	1	2000-11-11	2011-07-18
3	4	3	2007-11-18	
3	3	6	2004-11-27	
2	5	2	2017-08-24	2017-10-27

Delete

Back

(należy wpisać id_pracownika, id_klienta oraz id_książki oraz wypełnić datę oddanie następnie nacisnąć przycisk Update który zmieni wartość null podanego wiersza na podaną wartość. Aby klient mógł kolejny raz wypożyczyć tą samą książkę należy usunąć jego poprzedni wpis z datą oddania i wypożyczenia)













5. Dokumentacja:

Wprowadzanie danych:

Ze strony Klienta część danych takie jak podawanie tytułu jest generowane poprzez podanie właściwego tytułu id książki jest odnajdywane poprzez podany tytuł. I wtedy dany rekord jest dodawany do tablicy z Rezerwacji.

Wprowadzanie ze strony Pracownika jest ręczne może on wprowadzić dane i usunąć dane z każdej tabeli. Pod warunkiem że nie koliduje to z ważnymi informacjami, które powinny zostać w systemie.

Jak korzystać z aplikacji jest opisane w punkcie (4. Projekt funkcjonalności).

	checker-qual-3.5.0	28.12.2021 18:41	Executable Jar File	210 KB
	javafx-base-17.0.0.1	17.01.2022 18:34	Executable Jar File	1 KB
	javafx-base-17.0.0.1-win	17.01.2022 18:34	Executable Jar File	729 KB
	javafx-controls-17.0.0.1	17.01.2022 18:34	Executable Jar File	1 KB
	javafx-controls-17.0.0.1-win	17.01.2022 18:34	Executable Jar File	2 486 KB
	javafx-fxml-17.0.0.1	17.01.2022 18:34	Executable Jar File	1 KB
	javafx-fxml-17.0.0.1-win	17.01.2022 18:34	Executable Jar File	127 KB
	javafx-graphics-17.0.0.1	17.01.2022 18:34	Executable Jar File	1 KB
	javafx-graphics-17.0.0.1-win	17.01.2022 18:34	Executable Jar File	5 632 KB
	postgresql-42.3.1	28.12.2021 18:41	Executable Jar File	992 KB
	PostgreSQLJavaFX	17.01.2022 18:43	Executable Jar File	98 KB
	PostgreSQLJavaFX_jar	17.01.2022 20:14	Archiwum WinRA...	9 166 KB

Aby uruchomić aplikację należy rozpakować projekt i uruchomić PostgreSQLJavaFX jako plik typu Jar.

Lektury:

- <https://www.postgresql.org/>
- <https://www.w3schools.com/>
- <https://openjfx.io/>
- <https://javastart.pl/baza-wiedzy/java-ee/jdbc-podstawy-pracy-z-baza-danych>