# Bazy Danych 1 System zarzadzania biblioteka

Wiktor Urban

25 stycznia 2022

## 1 Projekt koncepcji, założenia

### 1.1 Temat projektu

Tematem projektu jest zrobienie aplikacji do zarzadzaniem biblioteka. W systemie bedzie można przegladać, dodawać, edytować i usuwać ksiażki, czytelników, bibliotekarzy. Bibliotekarz beda mogli wypożyczyć ksiażke czytelnikowi i ja przyjać. Jeśli ksiażka bedzie oddana po okresie wypożyczenia bibliotekarz przyjmuje opłate od czytelnika.

### 1.2 Analiza wymagań użytkownika

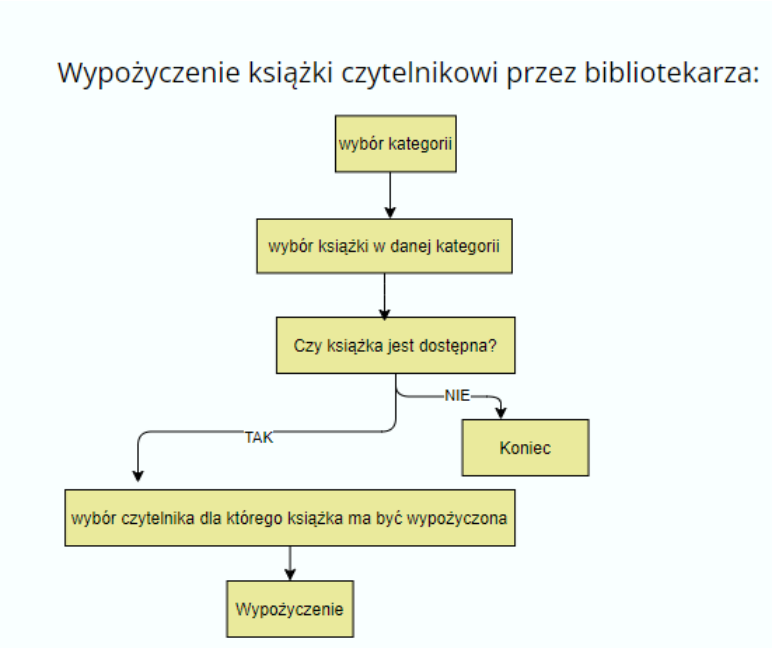
Czytelnik bedzie mógł wyszukiwać ksiażki po kategorii. Ksiażka bedzie mogła być wypożyczona o ile jest w bibliotece i jest dostepna (nie wypożyczona przez kogoś innego). Czytelnik może przegladać wypożyczone wcześniej przez niego ksiażki. Bibliotekarz może dodawać, usuwać i przegladać czytelników.

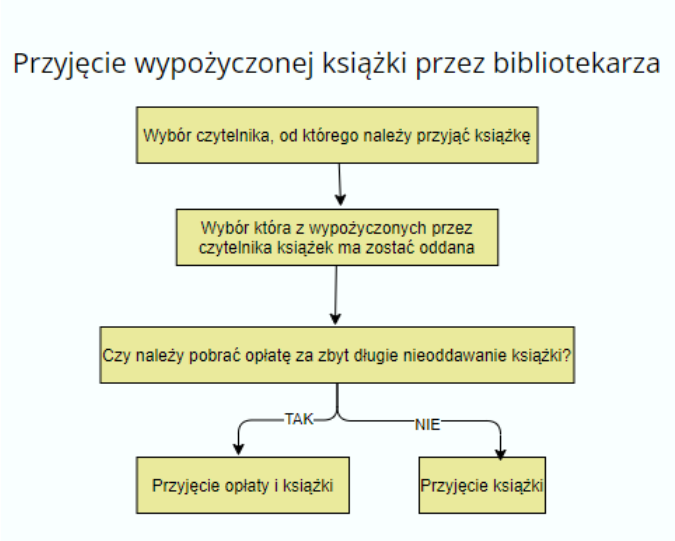
### 1.3 Zaprojektowanie funkcji

Dodanie i usuwanie czytelnika. Dodanie bibliotekarza. Wypożyczenie i oddanie ksiażki. Przyjecie opłaty za ksiażke oddana po terminie. Przegladanie kategorii ksiażek. Przegladanie ksiażek w danej kategorii. Przegladanie bibliotekarzy i czytelników. Przegladanie ksiażek wypożyczonych przez czytelnika.

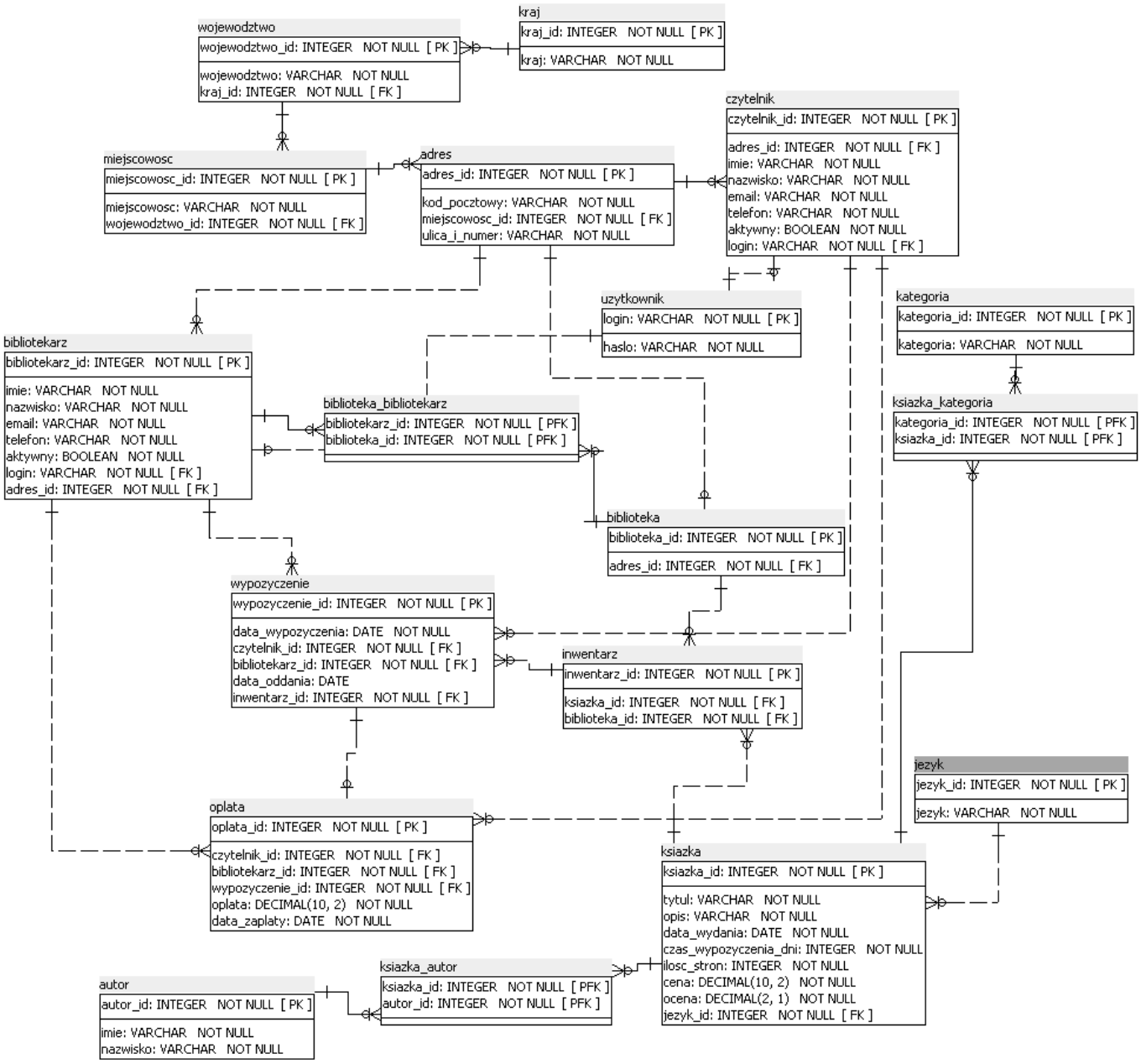
# 2 Projekt diagramów

## 2.1 Diagram przepływu danych



## 2.2 Diagram ERD



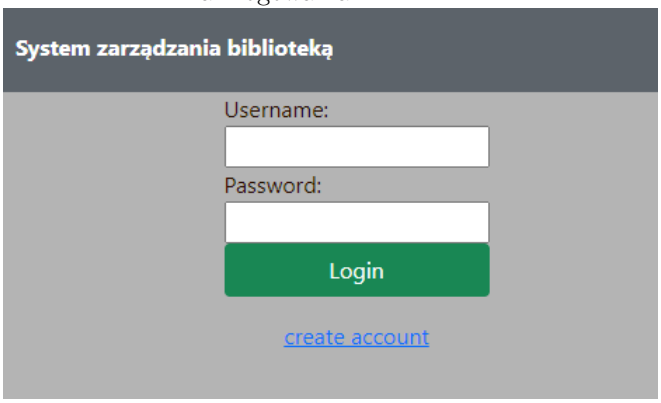

# 3 Projekt logiczny

Polecenia służace do zbudowania bazy danych wraz z danymi umieszczone sa w załaczonym pliku o nazwie dump.txt. Baze danych można również odbudować z pliku o nazwie backup.sql. Poza tablicami w bazie danych jest jeden widok 'ksiazka info' zbudowany na podstawie kilku tabel.

```sql
-- widok ksiazka info:
create view ksiazkaInfo as
select
    (select count(*) from wypozyczenie
        where inwentarz_id = inwentarz.inwentarz_id
            and data_oddania is null
    ) as niedostepna,
    kategoria_id, ksiazka_id, tytul, opis, data_wydania,
    czas_wypozyczenia_dni, ilosc_stron,
    cena, ocena, imie, nazwisko, biblioteka_id from ksiazka
            join ksiazka_kategoria using(ksiazka_id)
            join ksiazka_autor using(ksiazka_id)
            join autor using(autor_id)
            full outer join inwentarz using(ksiazka_id)
```

# 4 Projekt funkcjonalny

Ekran logowania




---

Widok po zalogowaniu. Użytkownik może być jednocześnie bibliotekarzem i czytelnikiem.

---

Przegladanie kategorii

**System zarządzania biblioteką** **Witaj testuser3** Twoje role: *Bibliotekarz Czytelnik*

- dodaj czytelnika
- przyjmij wypozyczona ksiazke
- Książki wg kategorii
- Moje wypozyczenia
- Bibliotekarze
- Czytelnicy
- wyloguj

kategorie i ilosc ksiazek w danej kategorii:

- Arts, 1
- Comics, 5
- Crafts, 1
- Entertainment, 4
- Fitness, 2
- Health, 3
- Hobbies, 1
- Kids, 2
- Math, 2
- Mysteries, 3
- Romance, 2
- Sciences, 3
- Sports, 5
- Travel, 1
- Biographies, 4
- Computers, 4
- Crime, 4
- Fantasy, 1
- Garden, 4
- Help, 2
- Home, 1
- Lesbian, 2
- Medical, 1
- Reference, 3
- Sci, 5
- Self, 2
- Tech, 2
- True, 5
- Business, 7
- Cooking, 3
- Edu, 1
- Fiction, 2
- Gay, 4
- History, 2
- Horror, 1
- Literature, 1
- Music, 4
- Religion, 1
- Science, 5
- Social, 4
- Teen, 4
- Westerns, 5

---

Przegladanie ksiażek w kategorii. Jeśli ksiażka jest niedostepna nie można jej wypożyczyć jednak dalej można zobaczyć szczegółowe informacje dotyczace ksiażki.

**System zarządzania biblioteką** **Witaj testuser3** Twoje role: *Bibliotekarz Czytelnik*

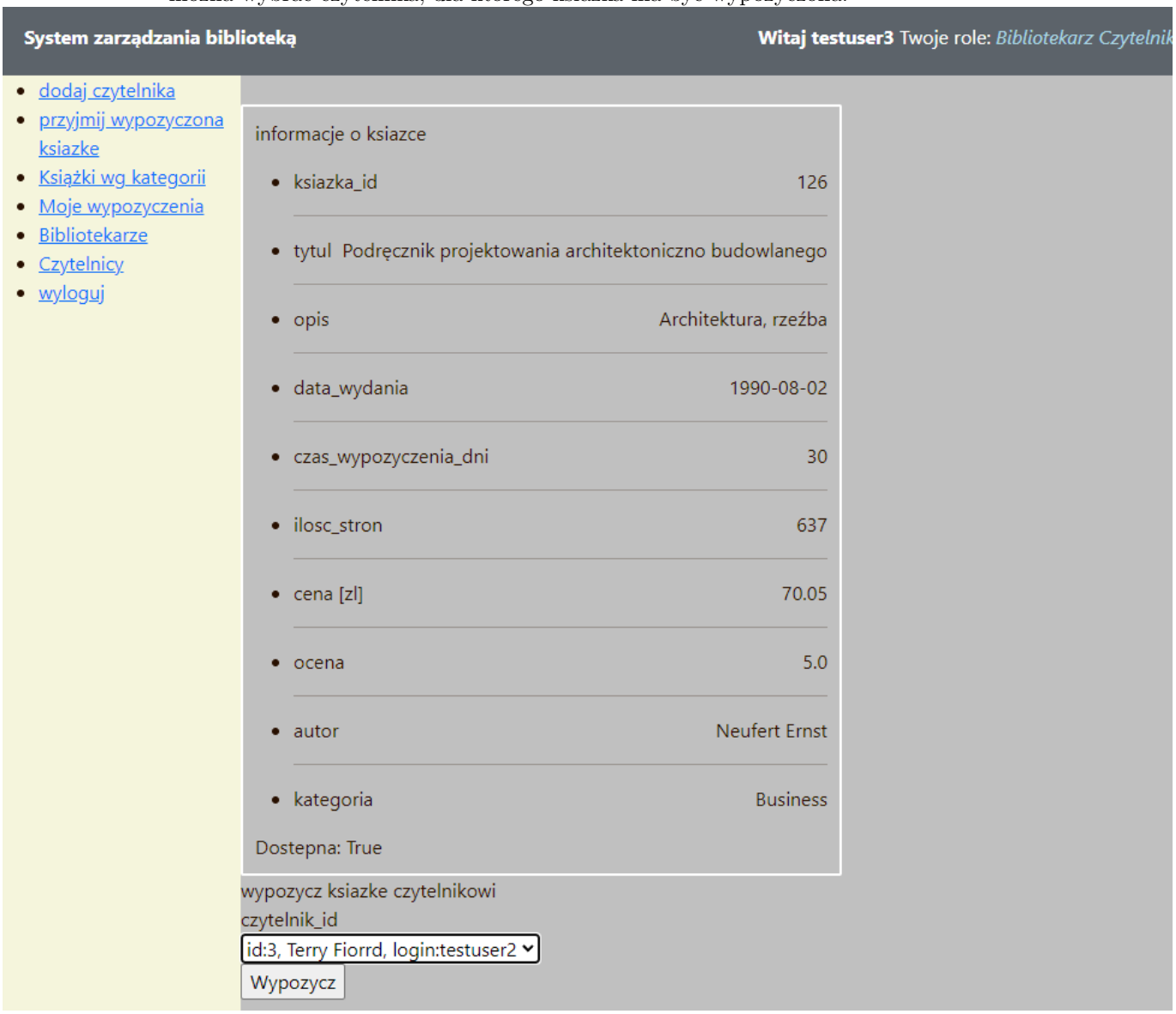
- dodaj czytelnika
- przyjmij wypozyczona ksiazke
- Książki wg kategorii
- Moje wypozyczenia
- Bibliotekarze
- Czytelnicy
- wyloguj

wyswietlanie ksiazek z kategorii Business

| id | Tytul | cena [zł] | autor | dostepnosc | wypozycz uzytkownikowi |
|---|---|---|---|---|---|
| 126 | Podręcznik projektowania architektoniczno budowlanego | 70.05 | Neufert Ernst | **dostepna** | wypozycz |
| 147 | Józef Mehoffer. Geniusz polskiej secesji | 52.95 | Luba Ristujczina | **niedostepna** | szczegoly |
| 155 | Malarstwo biało-czerwone Tom 2 | 145.32 | Waldemar Łysiak | **dostepna** | wypozycz |
| 237 | Muzeum Narodowe w Krakowie i Kolekcja Książąt Czartoryskich | 140.69 | Arkady no-surname | **dostepna** | wypozycz |
| 136 | Świat wnętrz Elle Decoration | 70.33 | Małgorzata Szczepańska | **dostepna** | wypozycz |
| 239 | Skarby UNESCO | 7.67 | Fenix no-surname | **dostepna** | wypozycz |
| 223 | Kazimierz Mikulski | 87.15 | Urszula Czartoryska | **dostepna** | wypozycz |

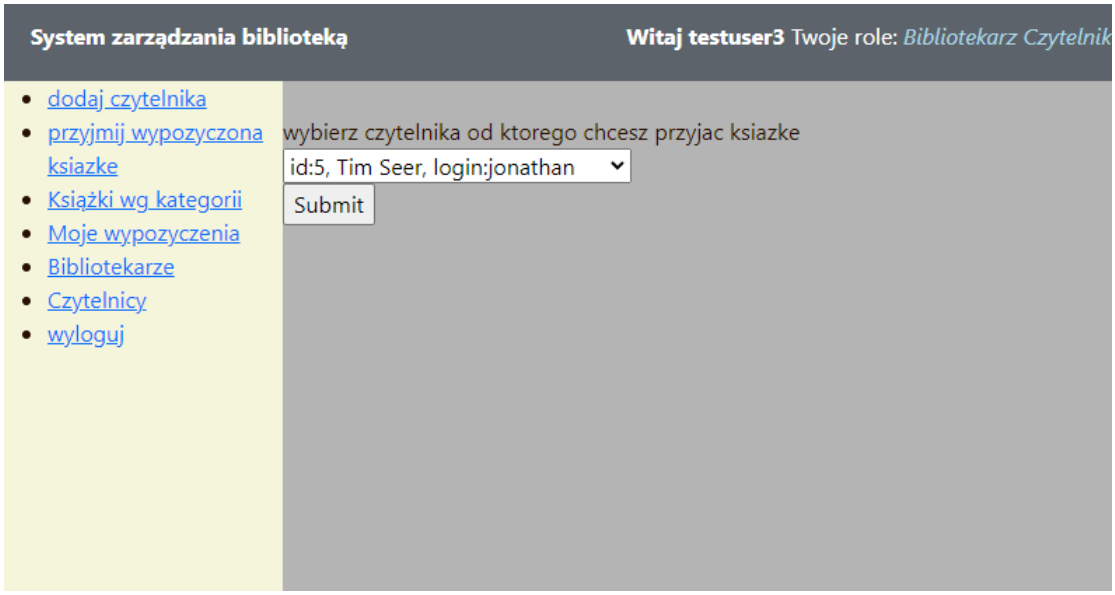
---

Widok wypożyczania ksiażki. Dostepne sa informacje na temat ksiażki oraz dropdown w którym można wybrać czytelnika, dla którego ksiażka ma być wypożyczona.



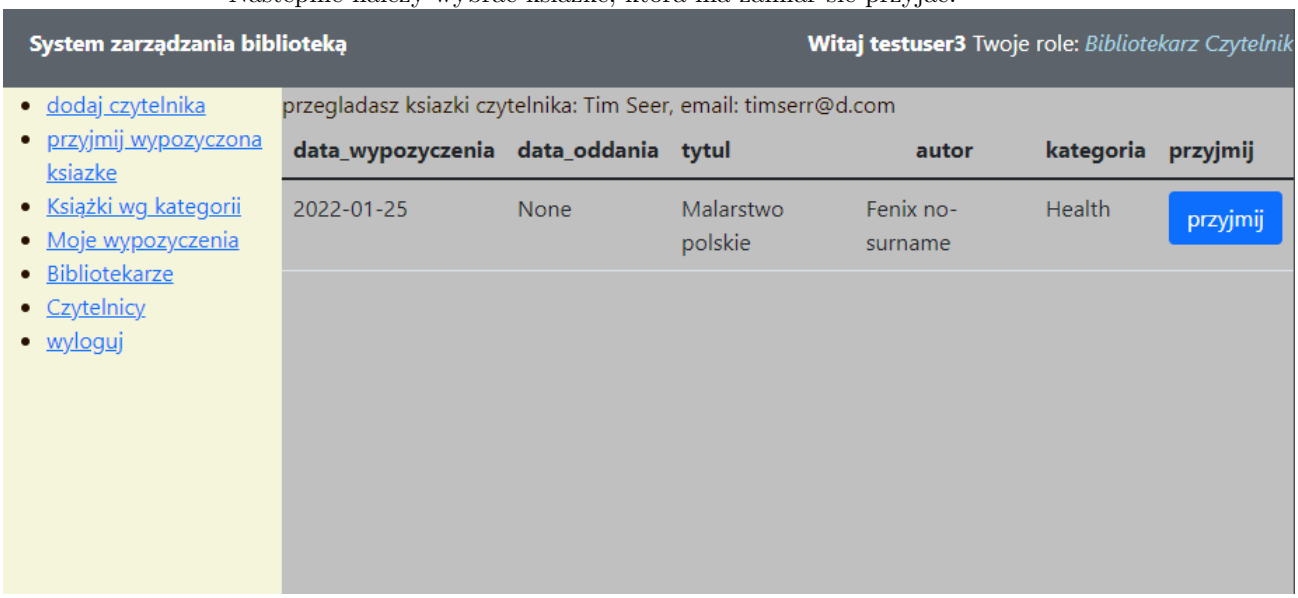

---

Przyjmowanie wypożyczony ksiażek od czytelnika. Najpierw należy wybrać czytelnika od którego chce sie przyjać ksiażke.



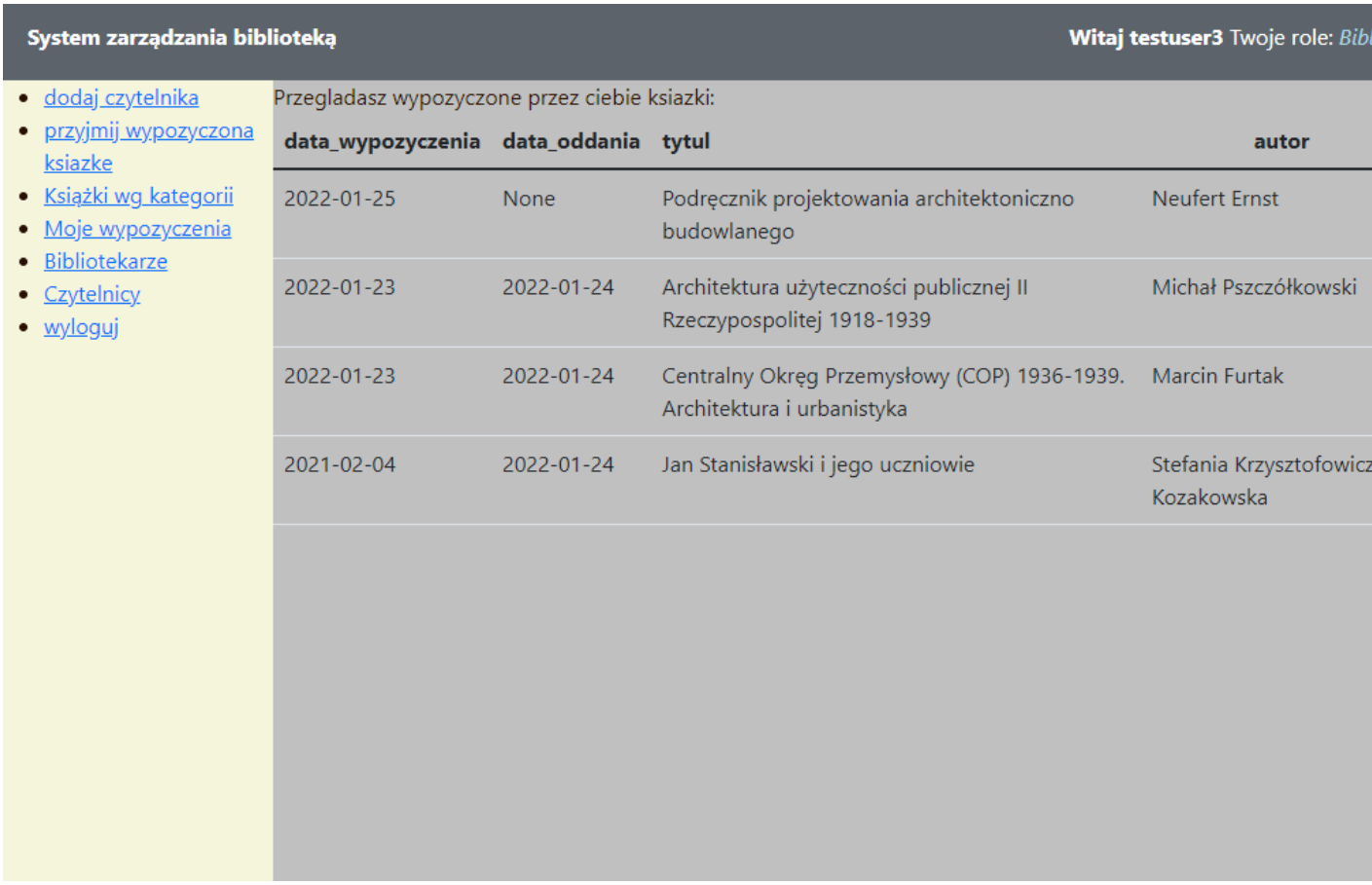

---

Nastepnie należy wybrać ksiażke, która ma zamiar sie przyjać.




---

Czytelnicy moga przegladać wypożyczone przez siebie książki.




# 5 Dokumentacja techniczna

Struktura folderów jest nastepujaca: libraryManagementSystem

- addLibrarian - funkcjonalność zwiazana z dodawaniem czytelników i bibliotekarzy
- library - główne funkcjonalności
- login - funkcjonalność zwiazana z logowaniem
- register - funkcjonalność zwiazana z rejestracja
- static - pliki css i scss
- readme - dokumentacja, erd, backup bazy danych i pgdump bazy danych

Ponadto pliki html umieszczone sa w folderach o nazwie ‘template‘.

Ustawienia zwiazane z routingiem umieszczone sa w plikach urls.py . Główny plik konfiguracyjny to plik urls.py w folderze wewnetrznym libraryManagementSystem.

Kod pythona odpowiedzialny za połaczenie z baza danych i ładowanie danych do strony jest w plikach views.py i commonFunctions.py. Projekt korzysta z pakietu psycopg2 do połaczenia z baza danych.

Cześć zapytań do bazy danych realizowana jest poprzez ‘prepare statement‘ aby zapobiec sql injection.

Przykładowy kod odpowiedzialny za wysłanie dokumentu html z tabela użytkowników:

```
def viewReaders(request):
    if not isUserLoggedIn(request): return redirect('/login/')
    context = {}
    addRolesToContext(request, context)
    context['readers'] = getReadersAsJson()
    template = loader.get_template('library/show-readers.html')
    return HttpResponse(template.render(context, request))

def getReadersAsJson():
    conn = connectToDB()
    cursor = conn.cursor()
    cursor.execute("
    select czytelnik_id, imie, nazwisko, email, aktywny from public.czytelnik;
    ")
    values = cursor.fetchall()
    valuesAsStringLists = convertTupleListToStringList(values)
    keys = ['czytelnik_id', 'imie', 'nazwisko', 'email', 'aktywny']
    jsonObj = jsonFromKeysAndStringLists(keys, valuesAsStringLists)
    conn.close()
    return jsonObj
```

# 6 Podsumowanie

Projekt spełnił założenia i zapewnia użytkownikom zadeklarowana funkcjonalność. Diagramy pomagaja w zrozumieniu struktury bazy danych, a załaczone pliki pozwalaja na zbudowanie bazy danych oraz aplikacji. Interfejs użytkownika pozwala na wykonanie zadeklarowanych działań.

# 7 Wykaz literatury


- https://sass-lang.com
- https://www.djangoproject.com
- https://www.python.org
- https://www.psycopg.org/docs/