

Bazy danych
Miniprojekt
Hibernate



AGH

Krzysztof Wójtowicz
Wojciech Sikora
Kacper Jankowski

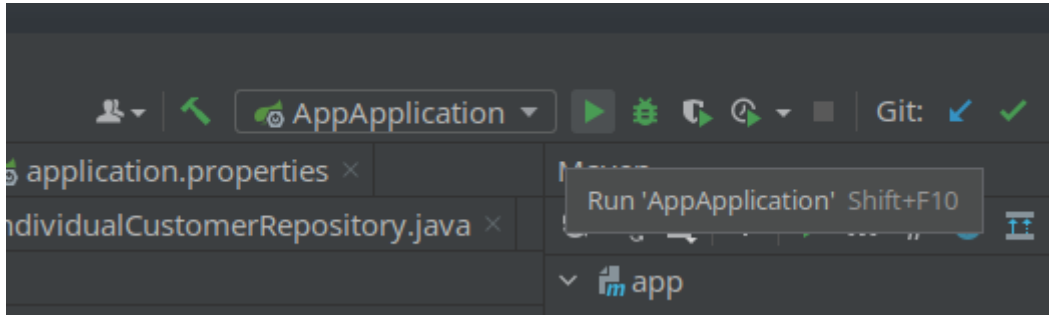
Link do projektu:

GitHub: <https://github.com/wojsikora/Database-project>

Sposób uruchomienia:

Z zainstalowanym Maven w terminalu w folderze bazowym wpisać "mvn spring-boot:run".

Używając GUI IntelliJ:



Opis:

W ramach projektu została zrealizowana aplikacja internetowa do obsługi usług wydruku 3D.

Zaimplementowane funkcje:

Poniżej wypisane zostały główne funkcjonalności aplikacji.

Z perspektywy klienta:

- rejestrowanie użytkowników, przechowywanie danych teleadresowych;
- składanie zamówień;
- załączanie pliku w formacie STL,
- specyfikowanie parametrów wydruku (m. in. grubość warstwy, rodzaj materiału, wypełnienie),
- zapis historii wydruków danego klienta

Z perspektywy osoby zarządzającej:

- zarządzanie treścią na stronie
- przegląd złożonych zamówień, możliwość zmian statusu zamówień;

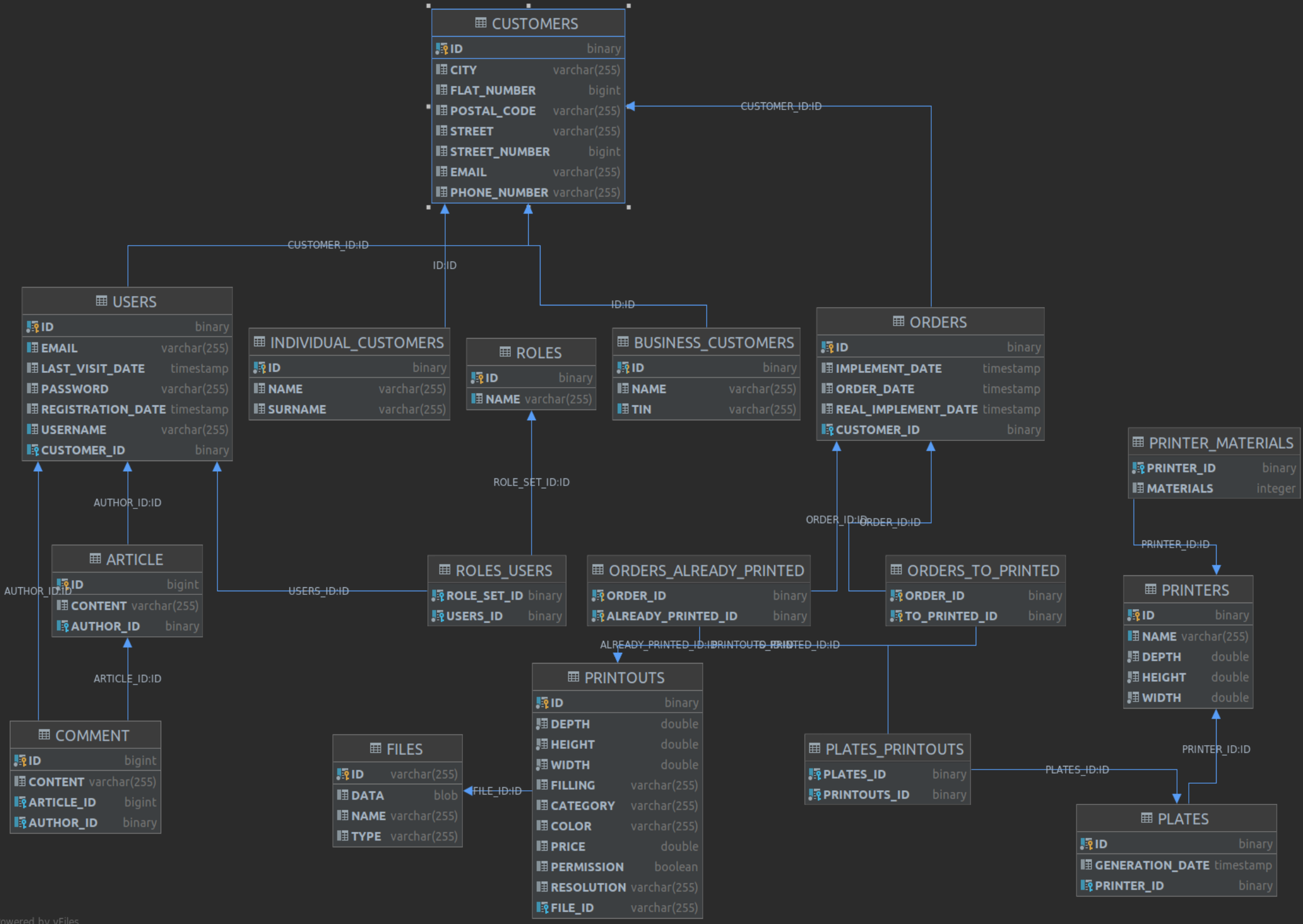
Zastosowane technologie:

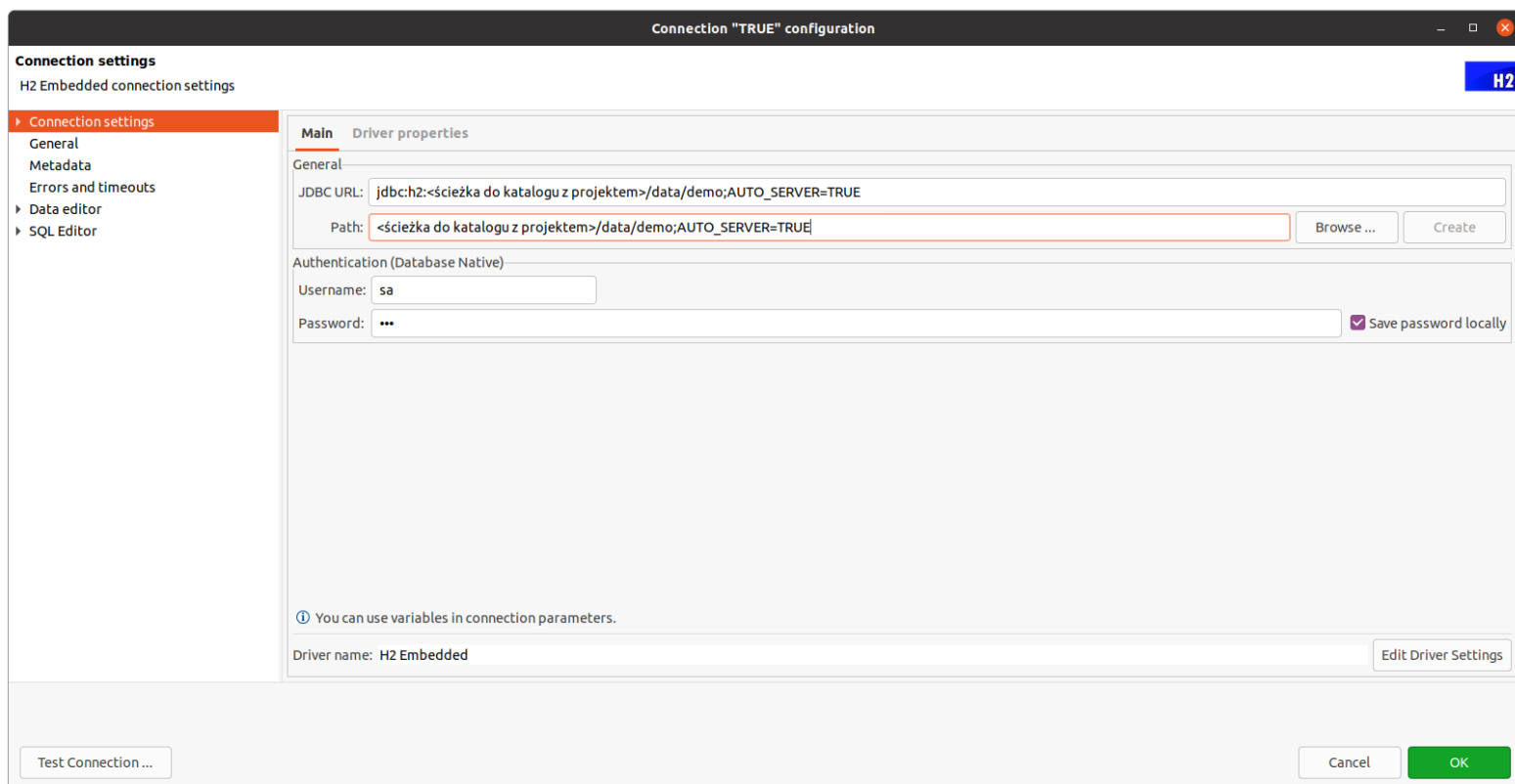
- Java
- Spring Boot
- **Spring HATEOAS**
- **Spring Data REST**
- **Spring Data JPA (Hibernate)**
- **Spring Validation**
- Spring Web
- Maven

Pogrubione elementy mają większe powiązanie z zarządzaniem danymi.

Wybraliśmy Javę z powodu naszego wcześniejszego zaznajomienia z tym językiem oraz MySQL z podobnych względów.

Chcemy zastosować pakiet Spring Boot, ponieważ jest szeroko stosowany w rynku IT, nadaje się do tego problemu i jest ukierunkowany na wspieranie tworzenia aplikacji internetowych korzystających z bazy danych. Udostępniane są w nim narzędzia, które zautomatyzują wiele procesów oraz zapewnią bezpieczeństwo realnych danych.





Zawartość bazy danych można podejrzeć łącząc się z nią przy pomocy dowolnego klienta bazodanowego używając powyższej konfiguracji (hasło dla użytkownika “sa” to “ppp”).

Struktura projektu została podzielona na cztery zasadnicze części:

- api - kontrolery obsługujące zapytania
- blog - część aplikacji odpowiedzialna za zarządzanie sekcją bloga,
- security - część aplikacji odpowiedzialna za zarządzanie użytkownikami i rolami
- order - główna część aplikacji, odpowiada za obsługę klientów i ich zamówień

Każda z sekcji blog, security oraz order została dodatkowo podzielona na trzy części:

- model - klasy modelujące dany obiekt
- repository - specjalne interfejsy odpowiedzialne bezpośrednio za komunikację z bazą danych, dostarczają funkcje odpowiedzialne za komunikację z bazą danych
- service - serwisy działające jako pomost pomiędzy metodami api a funkcjami dostępowymi do bazy danych

Zaimplementowane modele:

- a) **Model klient (Customer)** - model reprezentujący klienta. Zawiera id, e-mail, telefon oraz adres (zawierający miasto, kod pocztowy, ulicę, numer ulicy oraz numer mieszkania) klienta.
 - **indywidualny (IndividualCustomer)** - model reprezentujący klienta indywidualnego, jest podklasą klasy Customer. Zawiera imię i nazwisko klienta indywidualnego.
 - **biznesowy (BusinessCustomer)** - model reprezentujący klienta biznesowego, podklasa klasy Customer. Zawiera nazwę i numer NIP firmy
- b) **Model zamówienie (Order)** - model reprezentujący zamówienie. Zawiera id zamówienia, datę złożenia zamówienia, przewidywaną datę realizacji, datę realizacji, zbiór wydruków do wydrukowania, zbiór wydruków już zrealizowanych.
- c) **Model wydruk (Printout)** - model reprezentujący wydruk 3D. Zawiera id wydruku, wypełnienie, zgodę na wydruk, wymiary wydruku (reprezentowany przez 3 parametry - wysokość, szerokość, głębokość), materiał wydruku (zawierający kategorie materiału, kolor oraz cenę)
- d) **Model użytkownik (User)** - reprezentuje użytkownika. Zawiera id, nazwę, hasło, e-mail, datę rejestracji, datę ostatniego logowania, przysługujące użytkownikowi uprawnienia
- e) **Model drukarka (Printer)** - reprezentuje drukarkę. Zawiera id, nazwę, wymiary roboczego obszaru drukarki, kolekcję materiałów, którymi dana drukarka może drukować, kolekcję stołów z wydrukami
- f) **Model stół (Plate)** - reprezentuje zbiór modeli wydruków drukowanych jednorazowo w drukarce. Zawiera id, kolekcję wydruków, datę wydruku, przypisaną drukarkę.
- g) **Model artykułu (Article)** - reprezentuje artykuł. Zawiera id, użytkownika będącego autorem artykułu oraz zawartość artykułu.
- h) **Model komentarza (Comment)** - reprezentuje komentarz. Zawiera id, użytkownika będącego autorem komentarza, zawartość komentarza oraz artykuł, do którego odnosi się komentarz.

Zapytania do bazy danych zostały stworzone w wykorzystaniu repozytoriów Java Persistence API. Tłumaczą one metody na podstawie nazwy na odpowiednie zapytania do bazy danych. Pozwoliło to pracować na wyższym poziomie abstrakcji, bez bezpośredniego odnoszenia się do struktury bazy danych. Uprościło to także proces tworzenia nowych zapytań.

Przy takim podejściu pewną trudność stanowi pobieranie danych z wielu tabel jednocześnie, dlatego niektóre, specyficzne zapytania zostały zapisane w sposób szeregowy (na przykład najpierw pobrano modele danego typu, a następnie na podstawie pobranych danych pobrano odpowiednie modele innego typu).

System oferuje zestaw endpointów pozwalających na manipulację danymi:

- POST
 - /customers/business - dodawanie klienta biznesowego
 - /customers/individual - dodawanie klienta indywidualnego
 - /users - dodawanie użytkownika
 - /printers - dodawanie drukarki
 - /plate - dodawanie nowego stołu roboczego (zestaw projektów do wydrukowania jednocześnie)
 - /printouts - dodawanie nowego wydruku
 - /order - dodawanie nowego zamówienia
- GET
 - /printouts - pobieranie wszystkich użytkowników
 - /order/filtered?param=value - pobieranie zamówień o danych parametrach
 - /customers/business - pobieranie wszystkich klientów biznesowy
 - /customers/individual - pobieranie wszystkich klientów indywidualnych
 - /order/not_realized - pobieranie wszystkich niezrealizowanych zamówień
 - /plates - pobieranie wszystkich stołów roboczych drukarek
 - /plates/{id} - pobieranie stołu roboczego o danym id
 - /printers - pobieranie wszystkich drukarek
 - /printers/{id}/dimensions - pobieranie wymiarów obszaru roboczego danej drukarki
 - /printouts - pobieranie wszystkich wydruków
 - /users - pobieranie wszystkich użytkowników
 - /users/role - pobieranie użytkowników pełniących daną rolę

W pliku zapytania_postman.txt w repozytorium na GitHub zostały przygotowane zapytania, którym można wywołać po zaimportowaniu ich do programu Postman.

Poniżej przedstawiono efekty wywołań przykładowych zapytań

