

# **BDBT**

## **PROJEKT APLIKACJI DO OBSŁUGI BAZY DANYCH**

**DOM KULTURY**

**Grupa C**

**Marcin Ruta**

**Michał Prądziński**

**prowadzący: mgr inż. Tomasz Mrozek**

## **Dom kultury – spis treści**

- 1. Wykorzystana technologia**
- 2. Funkcjonalność aplikacji**
  - 2.1 Perspektywa pracownika**
  - 2.2 Perspektywa administratora**
- 3. Bibliografia**

## 1. Zastosowane technologie

W naszym projekcie skorzystaliśmy z środowiska Eclipse IDE for Java Developers oraz z języka programowania Java w wersji JDK 11. W środowisku Eclipse stworzyliśmy Maven project, aby ułatwić nam pracę i proces budowania projektu we frameworku Spring.

Maven to narzędzie, które automatyzuje budowę oprogramowania tzn. po utworzeniu specyfikacji projektu w pliku pom.xml tworzy odpowiednią dla danego projektu strukturę katalogową. Umożliwia automatycznie pobranie części wymaganych repozytoriów.

Dzięki Mavenowi inicjujemy projekt spring boot wraz z jego elementami takimi jak: SSpring JDBC, SSpring MVC czy SSpring Thymeleaf.

Spring Boot z wbudowanym serwerem Tomcat odpowiada za możliwość zbudowania aplikacji na serwerze lokalnym.

W naszym projekcie klasa uruchomieniowa serwera wygląda następująco:

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class DomKultury {

    public static void main(String[] args) {
        SpringApplication.run(DomKultury.class, args);
    }

}
```

JDBC (Java DataBase Connectivity) to interfejs, który pozwala na połączenie między naszą bazą danych a aplikacją napisaną w Javie. Dzięki JDBC możemy wysłać zapytania do bazy danych i otrzymywać z nich odpowiednie dane.

Użyliśmy Bazy danych Oracle Database 18c Express Edition postawioną na serwerach lokalnych. Dla tej bazy kompatybilny był sterownik JDBC w wersji 8.

Aby nie pisać specyfikacji połączenia w kodzie (co może być niebezpieczne), stworzyliśmy plik application.properties. W nim zawarliśmy m.in. url bazy danych oraz username oraz hasło użytkownika w naszej bazie Oracle.

Plik application.properties wyglądał tak:

```
spring.datasource.url = jdbc:oracle:thin:@127.0.0.1:1521/XEPDB1
spring.datasource.username = MRUTA
spring.datasource.password = admin
logging.level.root = INFO
```

Spring MVC jest używany do zarządzania warstwami aplikacji webowej. Dzieli się na trzy warstwy: Controller, Model oraz View. Model odpowiada za przechowywanie informacji. Controller zaś jest odpowiedzialny za logikę aplikacji. W naszym wypadku klasa używana jako kontroler to ApplicationController (oznaczona jest adnotacją @Controller). Natomiast warstwa View przedstawia przekazane dane w odpowiednim formacie.

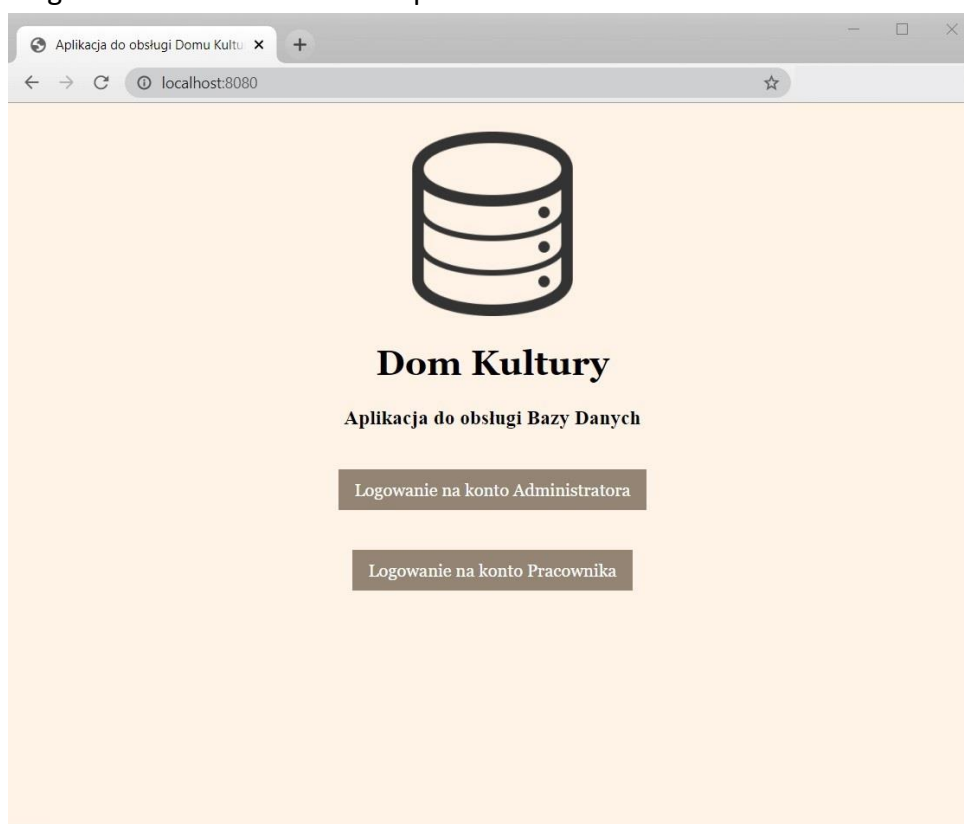
Thymeleaf jest to silnik szablonów, który jest zintegrowany ze Spring Bootem. Umożliwia proste i wygodne projektowanie formularzy. Odpowiada za połączenie warstwy modelu i widoku.

Dodatkowo skorzystaliśmy z testów jednostkowych (JUnitTest), aby sprawdzić czy napisaliśmy odpowiednio klasy w Javie, oraz z Oracle SQL Developer do kontroli danych w bazie – sprawdzania czy operacje CRUD (Create, Read, Update oraz Delete) wykonywane w aplikacji pokrywają się z rzeczywistymi danymi w Bazie danych.

Aby skoordynować naszą pracę (pracowaliśmy odlegle od siebie) używaliśmy repozytorium GitLab (wersję uczelnianą) do archiwizacji oraz wymiany plikami.

## 2. Funkcjonalność aplikacji

Strona startowa prezentuje się tak jak na obrazku poniżej. Dzięki niej możemy przejść do logowania administratora lub pracownika.



### 2.1 Perspektywa pracownika

Strona logowania pracownika:

Aplikacja do obsługi Domu Kultu x +

localhost:8080/LP

## Logowanie Pracownika

Użytkownik:	<input type="text" value="1"/>
Hasło:	<input type="password" value="*****"/>
<input type="button" value="Zaloguj się"/>	

Jeżeli wpisemy nieodpowiednie dane logowania wyskoczy nam błąd:

Aplikacja do obsługi Domu Kultu x +

localhost:8080/logErr

## Błąd logowania

Źle wprowadzono nazwę użytkownika lub hasło

Po poprawnym zalogowaniu się ujrzymy panel pracownika:

Aplikacja do obsługi Domu Kultu x +

localhost:8080/v/tables

## Tabele bazy danych - Dom Kultury

Tabela	Akcja
Twoje Domy Kultury	<input type="button" value="VIEW"/>
Twój adres	<input type="button" value="VIEW"/>
Twoje dane	<input type="button" value="VIEW"/>
Twoje wynagrodzenia	<input type="button" value="VIEW"/>
Obsługiwanie wydarzenia	<input type="button" value="VIEW"/>
Wydarzenia dostępne w ofercie Domu Kultury	<input type="button" value="VIEW"/>
Twoje realizacje wydarzeń wraz z Uczestnikami	<input type="button" value="VIEW"/>
Uczestnicy zapisani na twoje wydarzenia	<input type="button" value="VIEW"/>

Jeżeli klikniemy w przycisk „View” dla dowolnej tabeli wyświetlają się odpowiednie dane np. dla wiersza „Twoje dane” wyświetlają się dane danego pracownika:

ID	Imię	Nazwisko	Data urodzenia	PESEL	Nr Telefonu	Email	Płeć	Data Zatrudnienia	Data Zwolnienia	ID Domu	ID Adresu	ID Stanowiska
1	Stefan	Baczyński	1998-07-14	19980714123	123456789	stefanb123@gmail.com	Mężczyzna	2012-07-14		1	5	1

Powrót

Natomiast dla wiersza „Twój adres” strona wygląda następująco:

ID Adresu	Ulica	Nr Domu	Nr mieszkania	ID Poczty
5	Feliksa Nowowiejskiego	40		4

Powrót

Dla wiersza „Twoje realizacje wydarzeń wraz z Uczestnikami” strona internetowa ma dodatkową funkcję – jest możliwość wyświetlania uczestników dla danej realizacji przyporządkowanej dla zalogowanego pracownika.

ID Realizacji Wydarzenia	Data Wydarzenia	Cena	ID Wydarzenia	ID Sali	Uczestnicy
1	2020-12-11	10.5	1	1	<a href="#">Uczestnicy</a>

Powrót

Kiedy klikniemy na przycisk Uczestnicy, wyświetlą się, jak wcześniej było mówione, uczestnicy danej realizacji:



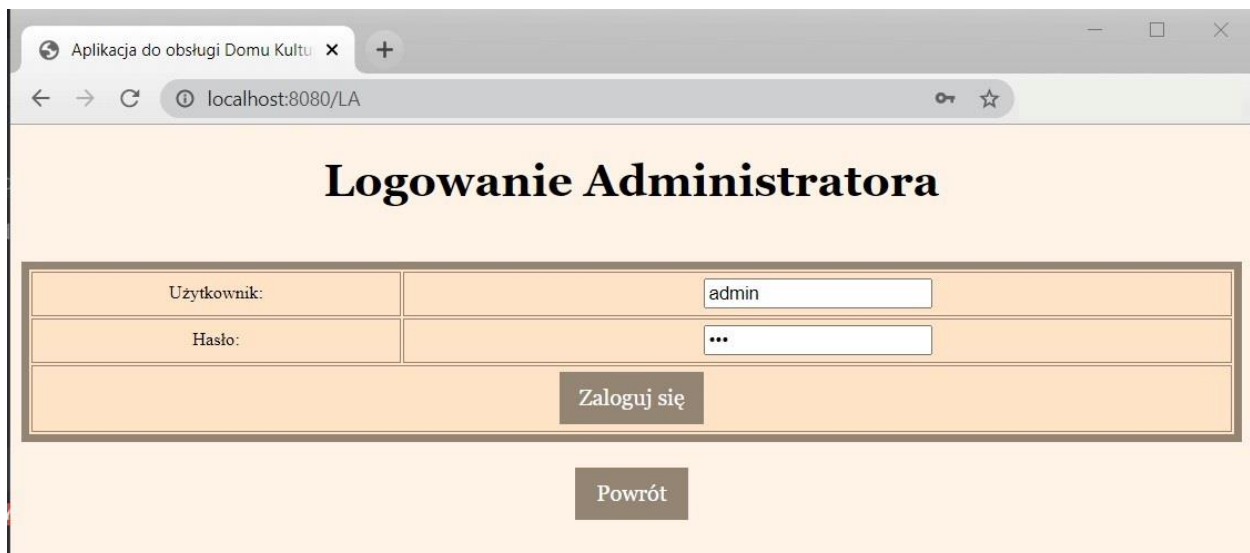
The screenshot shows a web browser window with the address bar displaying 'localhost:8080/v/RW/1'. The page title is 'Aplikacja do obsługi Domu Kultu'. The main heading is 'Uczestnicy danej realizacji'. Below the heading is a table with 8 columns: ID Uczestnika, Imie, Nazwisko, Plec, Data urodzenia, Numer Telefonu, Email, and ID Adresu. The table contains two rows of data. Below the table is a 'Powrót' button.

ID Uczestnika	Imie	Nazwisko	Plec	Data urodzenia	Numer Telefonu	Email	ID Adresu
1	Adam	Wojnarowski	Mezczyzna	2000-02-14		adamwojnarowski1@gmail.com	2
2	Michał	Wojnarowski	Mezczyzna	2000-02-14		michalwojnarowski53@gmail.com	2

Powrót

## 2.2 Perspektywa administratora

Logowanie administratora nie odbiega niczym szczególnym od logowania dla pracownika:



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/LA'. The page title is 'Aplikacja do obsługi Domu Kultu'. The main heading is 'Logowanie Administratora'. Below the heading is a login form with two input fields: 'Użytkownik:' with the value 'admin' and 'Hasło:' with a masked password '...'. Below the input fields is a 'Zaloguj się' button. At the bottom of the page is a 'Powrót' button.

Użytkownik: admin

Hasło: ...

Zaloguj się

Powrót

Jeżeli źle podamy dane logowania to, jak w przypadku logowania pracownika, wyświetli się nam strona z błędem. Jeżeli zaś poprawnie wpisujemy użytkownika i hasło to przejdziemy do widoku panelu administratora:

Aplikacja do obsługi Domu Kultu

+

localhost:8080/tables

🔑

☆

## Tabele bazy danych - Dom Kultury

Tabela	Akcja
Domy Kultury	VIEW
Adresy	VIEW
Pracownicy	VIEW
Wynagrodzenia	VIEW
Obsługiwanie wydarzenia	VIEW
Wydarzenia	VIEW
Realizacje wydarzeń	VIEW
Uczestnicy	VIEW
Zapisy na wydarzenie	VIEW

Wyloguj

Po kliknięciu w przycisk „View” wyświetla się odpowiednia tabela np. dla „Pracownicy” widok jest następujący:

Aplikacja do obsługi Domu Kultu

+

localhost:8080/P

☆

## Pracownicy Domu Kultury

Dodaj nowych Pracowników

ID	Imię	Nazwisko	Data urodzenia	PESEL	Nr Telefonu	Email	Płeć	Data Zatrudnienia	Data Zwolnienia	ID Domu	ID Adresu	ID Stanowiska	Akcja
1	Stefan	Baczyński	1998-07-14	19980714123	123456789	stefanb123@gmail.com	Mężczyzna	2012-07-14		1	5	1	Edit Delete
2	Emilia	Adamska	1997-08-23	19970823456	789456123	anitoalga@gmail.com	Kobieta	2013-07-06		1	6	2	Edit Delete
3	Micha	Dworak	1996-06-24	19960624456	598456123	kechamkocty@gmail.com	Mężczyzna	2014-06-14		1	4	3	Edit Delete
4	Kamil	Baczyński	1998-07-14			jkrfshjstjk@gmail.com	Mężczyzna	2013-02-18	2021-01-30	1	5	3	Edit Delete
5	Olga	Tuszyńska	1995-09-14			wojkte@gmail.com	Kobieta	2014-07-13		1	3	4	Edit Delete
43	asd	dsa	2005-05-05	12312312311	123123123	123@gmail.com	Mężczyzna	2005-05-05		1	1	1	Edit Delete
44	Papuzia	Papuzia	2000-05-05	12345678912	123456789	julka@gmail.com	Kobieta	2000-05-05	2021-01-30	1	2	1	Edit Delete
45	Marcin	Grunowski	1999-02-02	12345612345	123123123	123@gmail.com	Mężczyzna	1999-02-02		1	2	1	Edit Delete

Posortuj po ID

Posortuj po nazwisku

Powrót



W tym widoku można dodać nowych pracowników, zmodyfikować odpowiednie pola lub je usunąć. Jest też możliwość sortowania danych po ID oraz nazwisku. Naturalną rzeczą jest przycisk „Powrót”, który umożliwia przejście do panelu administratora.

Dodanie pracownika wygląda w następujący sposób:

The screenshot shows a web browser window with the title 'Aplikacja do obsługi Domu Kultu'. The address bar shows 'localhost:8080/newP'. The main heading is 'Wprowadz nowego Pracownika'. Below it is a form with the following fields and values:

Imię Pracownika:	test
Nazwisko Pracownika:	test
Data urodzenia:	14 . 01 . 2021
PESEL:	12345678911
NR Telefonu:	81238938
Email:	test@pw.edu.pl
Płeć:	Kobieta
Data Zatrudnienia:	dd . mm . rrrr
Data Zwolnienia:	dd . mm . rrrr Brak daty: <input checked="" type="checkbox"/>
ID adresu:	2
ID Domu:	1
ID Stanowiska:	3

At the bottom of the form is a 'Save' button. Below the form is a 'Powrót' button.

Jeżeli wpisujemy nieodpowiednie dane to nastąpi błąd (w powyższym przypadku nie wprowadziliśmy Daty Zatrudnienia, która nie może być nullem):

The screenshot shows a web browser window with the title 'Aplikacja do obsługi Domu Kultu'. The address bar shows 'localhost:8080/saveErr'. The main heading is 'Wprowadzono złe dane!'. At the bottom is a 'Powrót' button.

W przypadku poprawnego wprowadzenia danych nastąpi zapis formularza i przekierowanie na tabelę „Pracownicy”. Na poniższym screenie jest widoczny nowy pracownik - „test test” oraz dodatkowo posortowaliśmy widok ze względu na nazwisko.

Aplikacja do obsługi Domu Kultu

localhost:8080/RefreshPNazw

### Pracownicy Domu Kultury

Dodaj nowych Pracowników

ID	Imię	Nazwisko	Data urodzenia	PESEL	Nr Telefonu	Email	Płeć	Data Zatrudnienia	Data Zwolnienia	ID Domu	ID Adresu	ID Stanowiska	Akcja
2	Emila	Adamka	1997-08-23	19970823456	789456123	anitoalga@gmail.com	Kobieta	2013-07-06		1	6	2	Edit Delete
1	Stefan	Baczyński	1998-07-14	19980714123	123456789	stefanb123@gmail.com	Mężczyzna	2012-07-14		1	5	1	Edit Delete
4	Kamil	Baczyński	1998-07-14			jkrfihjsfjk@gmail.com	Mężczyzna	2013-02-18	2021-01-30	1	5	3	Edit Delete
43	asd	dsa	2005-05-05	12312312311	123123123	123@gmail.com	Mężczyzna	2005-05-05		1	1	1	Edit Delete
3	Micha	Dworak	1996-06-24	19960624456	598456123	kochamkocty@gmail.com	Mężczyzna	2014-06-14		1	4	3	Edit Delete
45	Marcin	Grunowski	1999-02-02	12345612345	123123123	123@gmail.com	Mężczyzna	1999-02-02		1	2	1	Edit Delete
44	Papuzia	Papuzia	2000-05-05	12345678912	123456789	julka@gmail.com	Kobieta	2000-05-05	2021-01-30	1	2	1	Edit Delete
61	test	test	2021-01-14	12345678911	81238938	test@pw.edu.pl	Kobieta	2021-01-14		1	2	3	Edit Delete
5	Olga	Tuszyńska	1995-09-14			wojte@gmail.com	Kobieta	2014-07-13		1	3	4	Edit Delete

Posortuj po ID

Posortuj po nazwisku

Powrót

Odpowiednie rekordy można usuwać, aczkolwiek zabronione jest usuwanie danych, które są powiązane z innymi tabelami. W takim przypadku wystąpi przekierowanie na odpowiednią stronę z błędem (w poniższym przykładzie chcieliśmy usunąć pracownika o ID = 2, który jest powiązany z tabelą wynagrodzenia):

Aplikacja do obsługi Domu Kultu

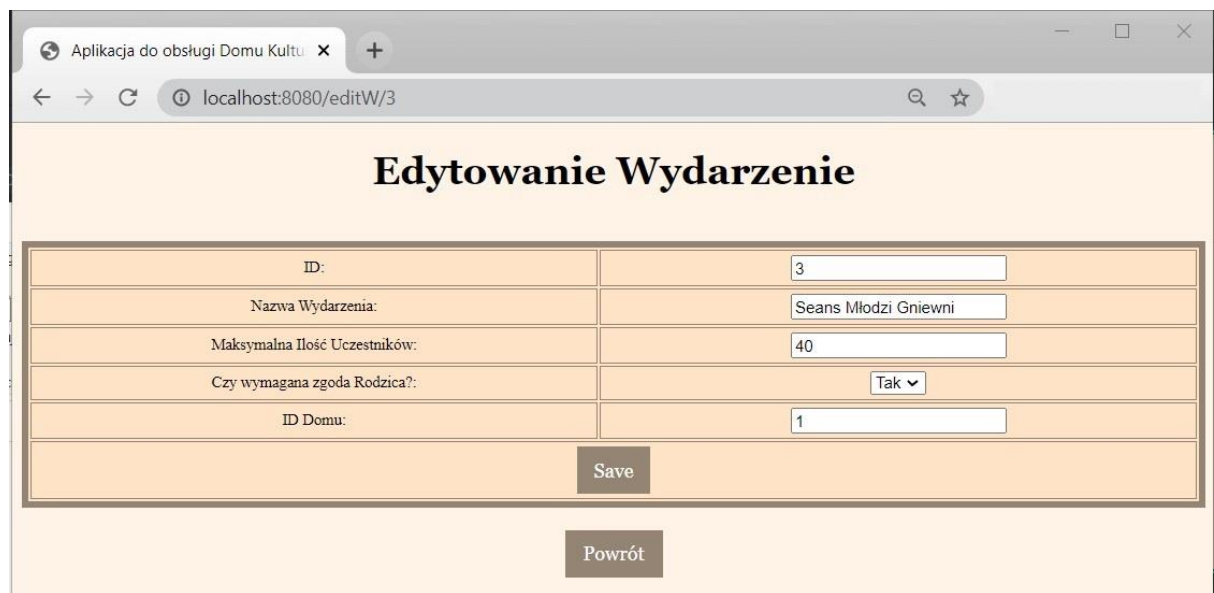
localhost:8080/delErr

## Nie można usunąć, ze względu na powiązania z innymi tabelami.

Aby usunąć należy najpierw usunąć związki pomiędzy danym rekordem a innymi tabelami.

Powrót

Edytowania rekordu pokażemy na przykładzie widoku Wydarzeń:



Aplikacja do obsługi Domu Kultu x +

localhost:8080/editW/3

## Edytowanie Wydarzenie

ID:	3
Nazwa Wydarzenia:	Seans Młodzi Gniewni
Maksymalna Ilość Uczestników:	40
Czy wymagana zgoda Rodzica?:	Tak ▼
ID Domu:	1

Save

Powrót

Po zmodyfikowaniu pól „Maksymalna Ilość Uczestników” oraz „Czy wymagana zgoda Rodzica?” odpowiednio na wartości „20” oraz „Nie” i naciśnięciu przycisku „Save” nastąpi zapisanie zmian oraz przekierowanie na widok tabeli „Wydarzenia”:



Aplikacja do obsługi Domu Kultu x +

localhost:8080/W

## Wydarzenia zapisane w Domu Kultury

Dodaj nowe Wydarzenie

ID Wydarzenia	Nazwa Wydarzenia	Maksymalna ilość uczestników	Wymagana zgoda rodzica	ID Domu	Akcja
1	Język Angielski	25	N	1	Edit Delete
2	Nauka Tańca Towarzyskiego	20	N	1	Edit Delete
3	Seans Młodzi Gniewni	20	N	1	Edit Delete
4	Język Hiszpański	25	T	1	Edit Delete

Powrót

### **3. Bibliografia**

**Tutorial html:**

<https://www.w3schools.com/html/>

**Tutorial css:**

<https://www.w3schools.com/css/default.asp>

**Generator tabel html:**

<https://divtable.com/table-styler/>

**Tutorial thymeleaf:**

<https://www.thymeleaf.org/doc/tutorials/2.1/usingthymeleaf.html>

**Tutorial Spring:**

<https://www.tutorialspoint.com/spring/index.htm>

**MVC:**

<https://www.javatpoint.com/spring-mvc-tutorial>