

Dokumentacja projekt BADA

Część 2

Jakub Kałuski, Szymon Miareczka,
Cyberbezpieczeństwo GR A

Politechnika Warszawska, Wydział Informatyki i Technik Informatycznych

22 maja 2021

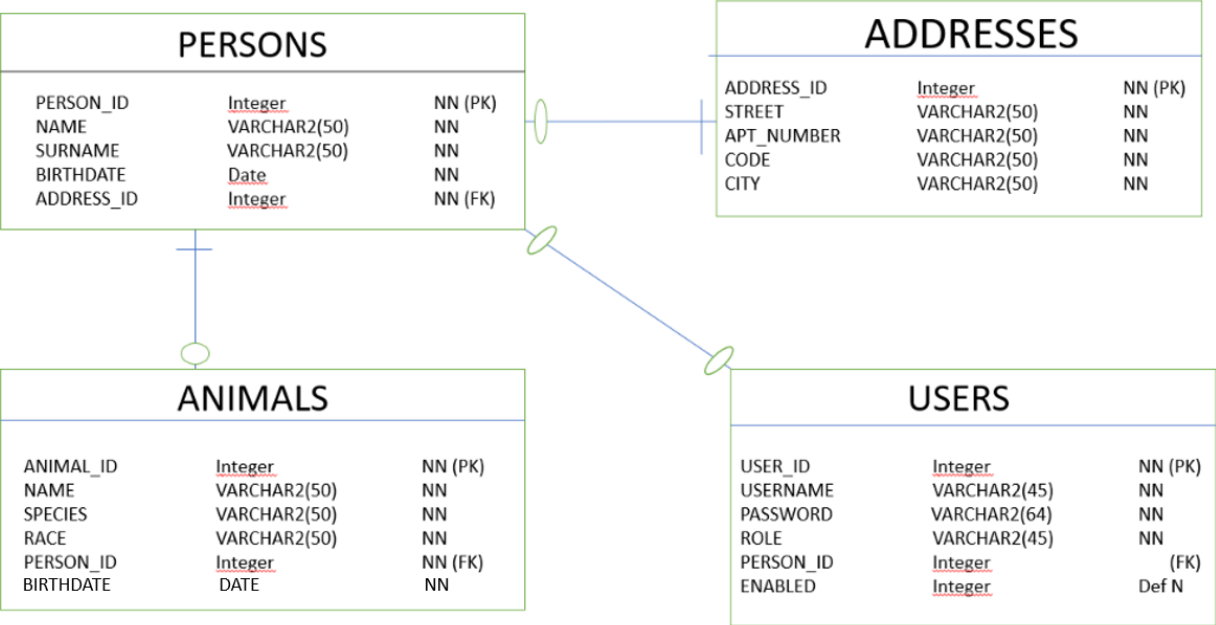
Spis treści

1	Zakres i cel projektu	1
2	Schemat przykładowej bazy danych	2
3	Opis wykorzystanej technologii	3
4	Możliwe operacje na tabelach PERSONS, ADDRESSES, ANIMALS	3
5	Perspektywy użytkowników	4
6	Zabezpieczenie przed błędami w bazie danych	4
7	Działanie aplikacji	5
7.1	Strona główna	5
7.2	Bez zalogowania	5
7.3	Logowanie	6
7.4	Wylogowanie	6
7.5	Widok admina	7
7.6	Widok Usera	7
7.7	Dodawanie nowych rekordów	8
7.8	Edycja rekordów	8
7.9	Obsługa błędów	9

1 Zakres i cel projektu

Celem projektu było przygotowanie aplikacji współpracującej z zaprojektowaną niewielką bazą danych.

2 Schemat przykładowej bazy danych



Rys. 1: Wykorzystana baza danych

3 Opis wykorzystanej technologii

- Język programistyczny java
- Język programistyczny HTML
- IntelliJ Idea
- SQL Developer
- Baza danych oracle w wersji 19c
- Java Spring
- Java Spring Security

Aby stworzyć aplikację spełniającą zakres i cel projektu skorzystaliśmy z 2 języków programowania Java i HTML.

Jeśli chodzi o środowisko programistyczne, to skorzystaliśmy z dobrze znanego nam IntelliJ IDEA firmy JetBrains. Korzystamy z tego środowiska od początku studiów. Uważamy je za bardzo intuicyjne i przyjemne w obsłudze. Stąd nasza decyzja.

Przy tworzeniu naszej aplikacji pomógł nam Apache Maven, czyli narzędzie automatyzujące budowę oprogramowania na platformę Java. Przy jego pomocy bez problemu mogliśmy skorzystać ze frameworka Spring, używanego przez duże firmy do tworzenia aplikacji biznesowych. Nasza aplikacja została napisana zgodnie z modelem MVC(Model-View-Controller).

Jeśli chodzi o bazę danych to skorzystaliśmy z wcześniej używanej na tym przedmiocie bazy danych Oracle w wersji 19c. Do nawiązania połączenia z bazą używaliśmy SQL Developer, czyli zintegrowanego środowiska programistycznego, pozwalającego na pracę z bazą danych Oracle. Do połączenia naszej aplikacji z Bazą danych użyliśmy interfejsu JDBC.

Zgodnie z założeniem stworzyliśmy również perspektywy użytkowników. By to zrobić skorzystaliśmy z Spring Security, obsługujące między innymi logowanie 4 i wylogowanie. 5

4 Możliwe operacje na tabelach PERSONS, ADDRESSES, ANIMALS

- przeglądanie danych
- usuwanie danych
- dodawanie danych
- modyfikowanie danych

5 Perspektywy użytkowników

Mamy 2 perspektywy:

- admin
- user

Na stronach jest podany nickname użytkownika zalogowanego.
W prawym górnym rogu jest przycisk do zalogowania lub wylogowania.

Admin - może wszystko. Ma dostęp do 3 tabel (PERSONS, ADDRESSES 6, ANIMALS). Może dodawać 8, przeglądać, usuwać i edytować dane.⁹

User - Zwykły użytkownik bazy danych. Ma dostęp tylko do swoich danych. Może je tylko przeglądać. ?? Dane usera są pobierane przez połączenie tablicy USERS z tablicą PERSONS. W tablicy USERS przechowujemy dane użytkownika oraz klucz obcy *PERSON_ID*, dzięki któremu możemy userowi przypisać odpowiednie dane. Do tablicy users można wprowadzać dane tylko z poziomu SQLDeveloper. W aplikacji widoczne są tylko 3 tabele (PERSONS, ADDRESSES, ANIMALS). Gdy użytkownik jest niezalogowany nie może nawet przeglądać żadnych danych. 3

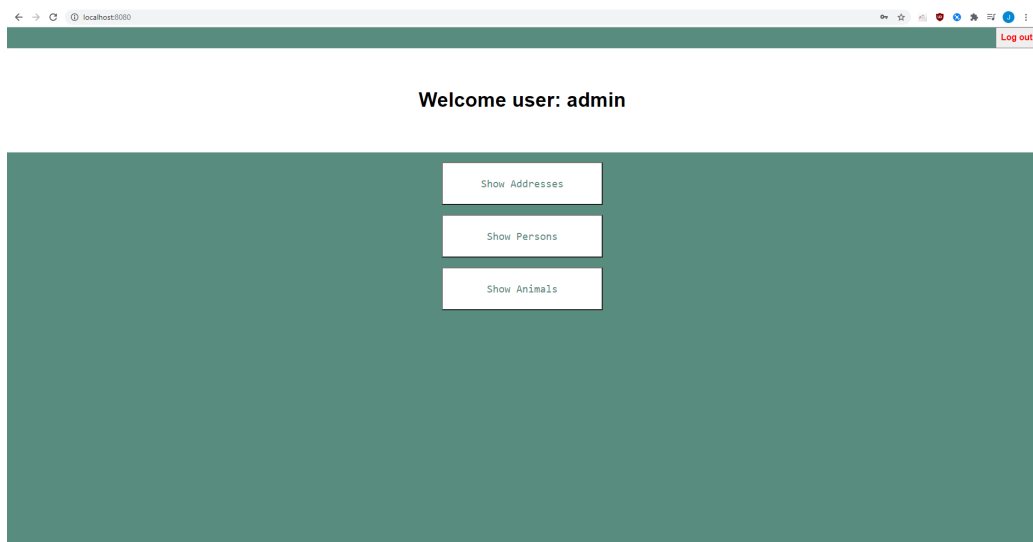
6 Zabezpieczenie przed błędami w bazie danych

Zabezpieczyliśmy naszą aplikację przed występowaniem błędów w bazie danych przez wysłanie złego query poprzez dodanie odpowiednich przekierowań,

- w przypadku, gdy usuwamy jakiś rekord, którego klucz podstawowy jest kluczem obcym jakiegoś rekordu w jakiejś tabeli. 12
- gdy próbujemy dodać wartości puste 10
- gdy dodajemy rekord z kluczem obcym, który nie ma pokrycia w kluczu własnym

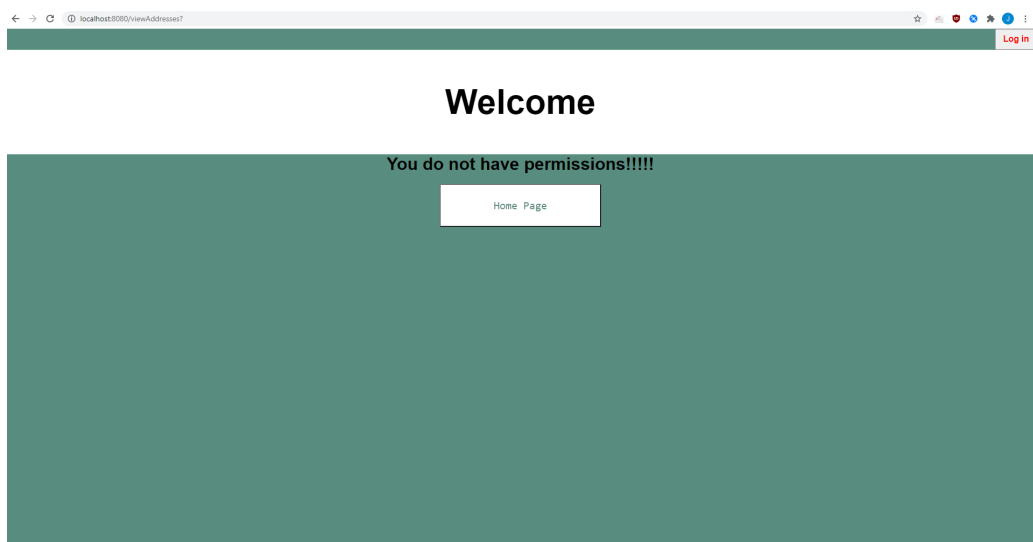
7 Działanie aplikacji

7.1 Strona główna



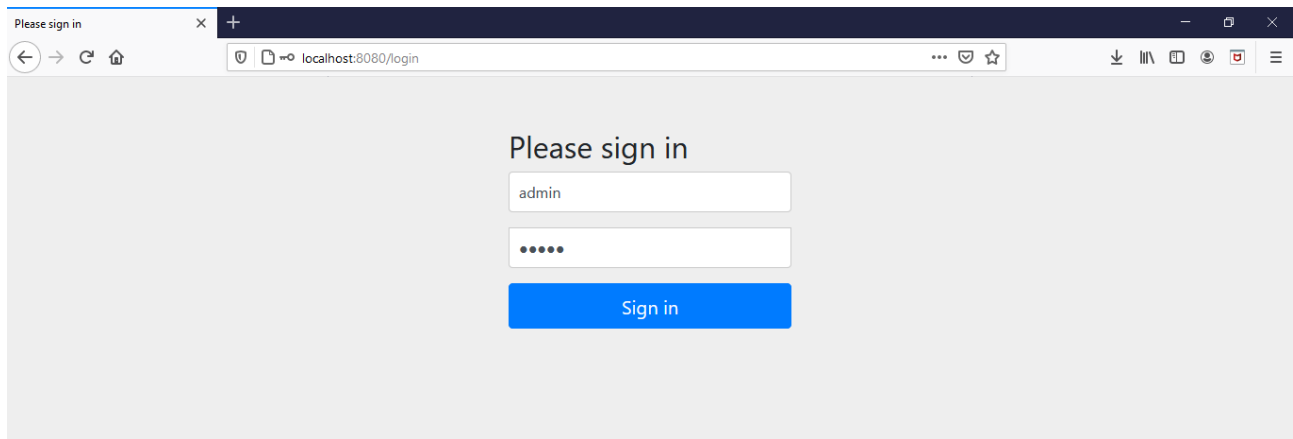
Rys. 2: Strona główna po logowaniu

7.2 Bez zalogowania



Rys. 3: Próba otworzenia rekordów przez niezalogowanego użytkownika

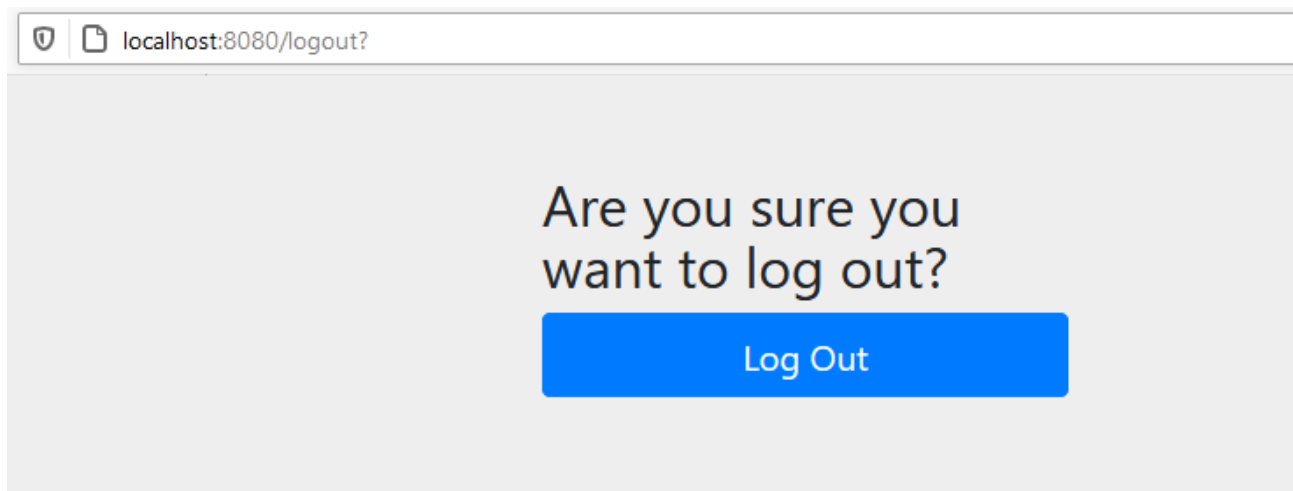
7.3 Logowanie



Rys. 4: Strona logowania

7.4 Wylogowanie

Następuje po użyciu przycisku log out z lewego górnego rogu



Rys. 5: Wylogowanie

7.5 Widok admina

Zawartość strony Addresses (analogicznie Persons i Animals) z perspektywy admina.

Log out					
user: admin					
ADDRESSES					
Enter new address					
ADDRESS_ID	STREET	APT_NUMBER	CODE	CITY	Action
1	Wyszynskiego	51	05-100	Płock	Edit Delete
2	Wyszynskiego	54	05-100	Warszawa	Edit Delete
28	Legionowska	3a	05-100	Warszawa	Edit Delete
29	Mickiewicza	24c	05-120	Legionowo	Edit Delete
Home Page					

Rys. 6: Adresy z bazy danych

7.6 Widok Usera

Ograniczone tylko do jego rekordów

Log out					
user: user3					
ADDRESSES					
Enter new address					
ADDRESS_ID	STREET	APT_NUMBER	CODE	CITY	Action
1	Wyszynskiego	51	05-100	Płock	Edit Delete
Home Page					

Rys. 7: Adres zalogowanego usera

7.7 Dodawanie nowych rekordów

Dostępne dla admina, po kliknięciu opcji Enter new Address (analogicznie Person, Animal)

Log

user: admin

Add new address

Street:	<input type="text"/>
Apt number:	<input type="text"/>
Code:	<input type="text"/>
City:	<input type="text"/>

Save

Home Page

Rys. 8: Strona dodawania nowego adresu

7.8 Edycja rekordów

Dostępne dla admina, po wciśnięciu Edit book widocznego rekordu, brak możliwości zmiany klu-
cza podstawowego.

Analogicznie dla Animals i Persons

Log

user: admin

Edit address

ADDRESS_ID:	<input type="text" value="1"/>
STREET:	<input type="text" value="Polonistyczna"/>
APT_NUMBER:	<input type="text" value="26"/>
CODE:	<input type="text" value="20-604"/>
CITY:	<input type="text" value="Gdańsk"/>

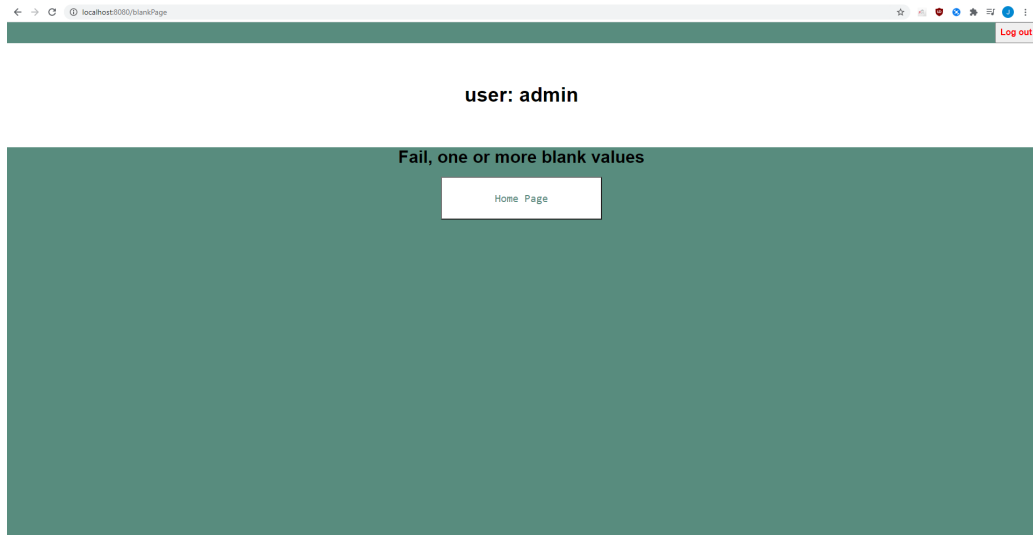
Save

Home Page

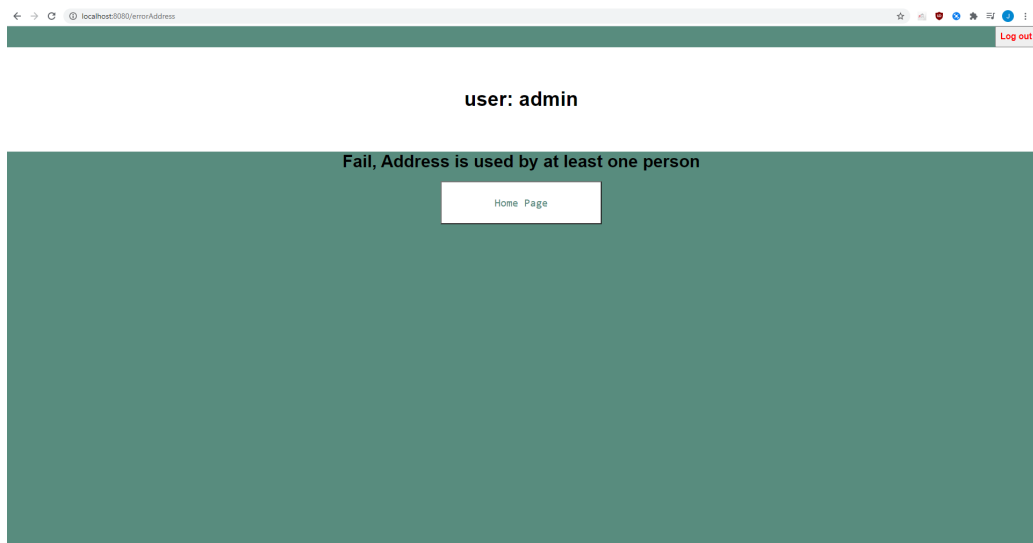
Rys. 9: Edycja istniejącego adresu

7.9 Obsługa błędów

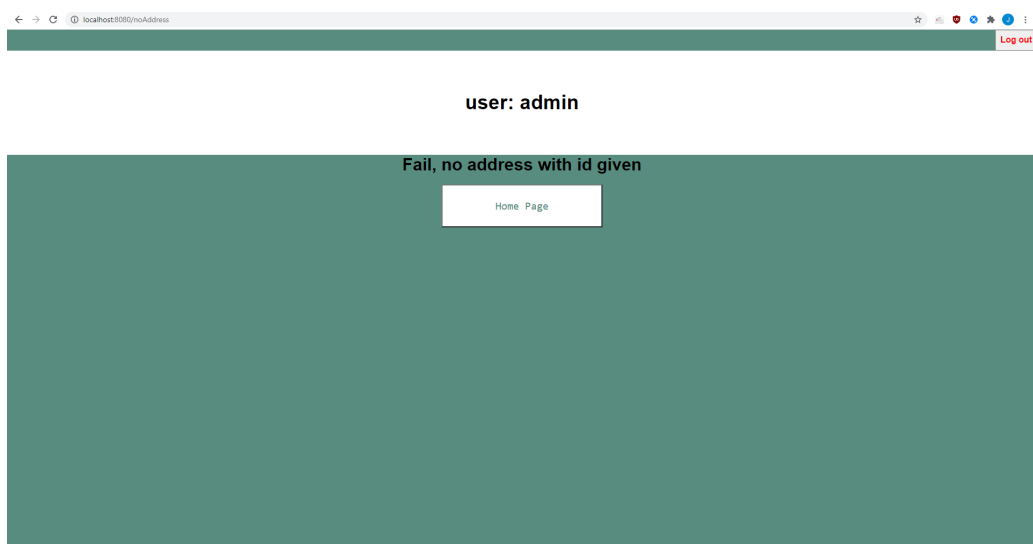
Próby usunięcia (po wciśnięciu Delete obok rekordu) związane z naruszeniem więzów integralności



Rys. 10: Próba usunięcia osoby, która ma jakieś zwierzęta



Rys. 11: Próba usunięcia adresu, który posiada jakaś osoba



Rys. 12: Próba dodania osoby z polem ADDRESS_ID, którego nie ma w tabeli ADDRESSES