DOKUMENTACJA TECHNICZNA

Nazwa projektu:

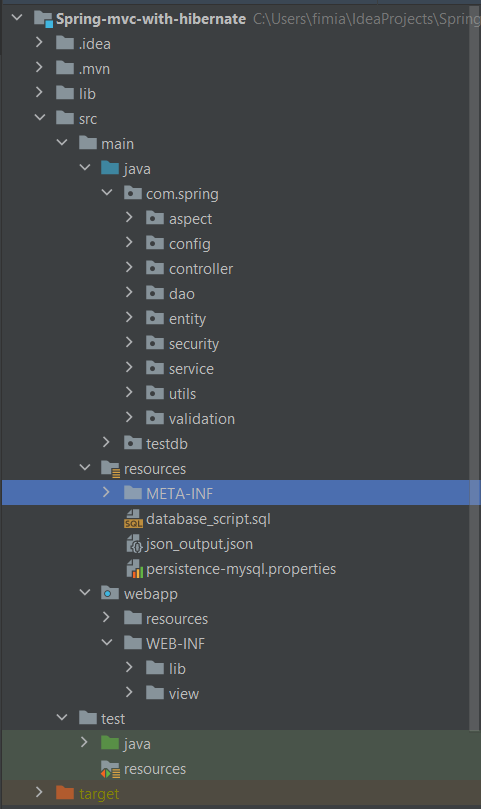
Web Issue Tracker

Biblioteki i technologie wymagane:

* Windows
* MySQL.
* Java 11.0.15.1
* Maven

Do zainstalowania potrzebna jest baza utworzona przez plik „database\_script.sql” oraz klon projektu.

Drzewo projektu:



1. Pakiety w folderze /src/main/java:

1.Pakiet „config”:

Odpowiada za konfiguracje aplikacji, połączenie z bazą, ustawienie opcji frameworków spring, spring mvc, hibernate oraz innych potrzebnych ustawień. Klasa CustomAuthenticationSuccessHandler za obsługę poprawnego logowania, ustawienia sesji użytkownika i przekierowanie go na stronę główną.

2.Pakiet „security”:

Konfiguracja spring security, zdefiniowanie konfigu aplikacji oraz zabezpieczenia konkretnych widoków dla użytkowników bez danych ról. Odpowiada też za szyfrowanie haseł.

3.Pakiet „entity”:

Klasy entity framework`u hibernate. Pola w klasach odpowiadają kolumnom w bazie danych. Mają tylko konstruktory, gettery i setery. Klasa CRMUser odpowiada za uwierzytelnianie użytkowników, sama nie ma reprezentacji w bazie, po zarejestrowaniu użytkownika wartości pól przepisywane są do klasy User.

4.Pakiet „dao”:

Obiekty dostępu do danych. Obsługują porozumiewanie się z bazą, wykonują zapytania i zwracają rezultat. Dzielą się na interfejs i implementację. Np.: Klasa IssueDAO definiuje metody typu: getIssue(), getIssues(), deleteIssue()… które muszą zostać zaimplementowane przez klasę IssueDAOImpl.

5.Pakiet „service”:

Pomocnicze klasy stanowiące fasadę pomiędzy DAO a kontrolerem w wypadku, gdy w przyszłości będzie potrzeba zaimplementowania bardziej skomplikowanej logiki/skorzystania z kilku DAO przy jednej metodzie.

6.Pakiet „aspect”:

Klasa obsługująca wszelkie problemy cross-cutting które wymagają wpływu na całą aplikacje. W tej chwili wykorzystywana do zapisywania logów w konsoli.

7.Pakiet „validation”:

Walidacja poprawnie wpisanych danych podczas rejestarcji.

8.Pakiet „utlis”:

Dowolne klasy pomocniczne.

2. Folder src/main/resources:

1.META-INF:

Metadane mapowania obiektowo-relacyjnego presistance.

2.Pozostałe pliki:

database\_script.sql - plik tworzący bazę, tabele i zależności.

json\_output.json – output dumpowania bazy do pliku json

persistance-mysql.properties – konfiguracja połączenia z bazą danych, podstawowe ustawienia połączenia i hibernate.

3. Folder src/main/webapp:

1.resources: pliki css opisujące wygląd niektórych aspektów aplikacji.

2.view: widoki modelu MVC, wszystkie widoki aplikacji w formacie .jsp.

4. Folder src/test:

Standardowy, utworzony przez maven folder do pisania testów jednostkowych.

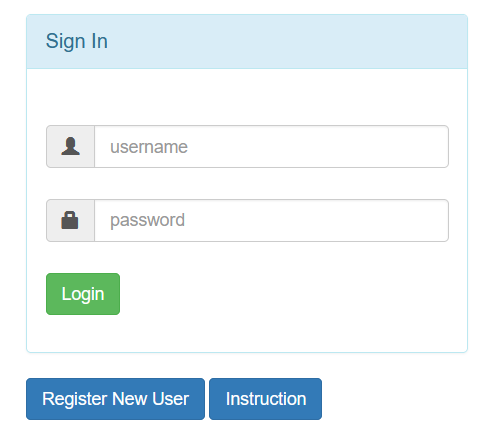
5.REST API:

Operacje CRUD z wykorzystaniem REST API:

|  |  |  |
| --- | --- | --- |
| METODA | OPERACJA | API |
| GET | Wyświetl wszystkie wpisy | /issues |
| GET | Wyświetl wpis o id | /issues/{issueId} |
| POST | Stwórz wpis | /issues |
| PUT | Edytuj wpis | /issues |
| DELETE | Usuń wpis | /issues/{issueId} |

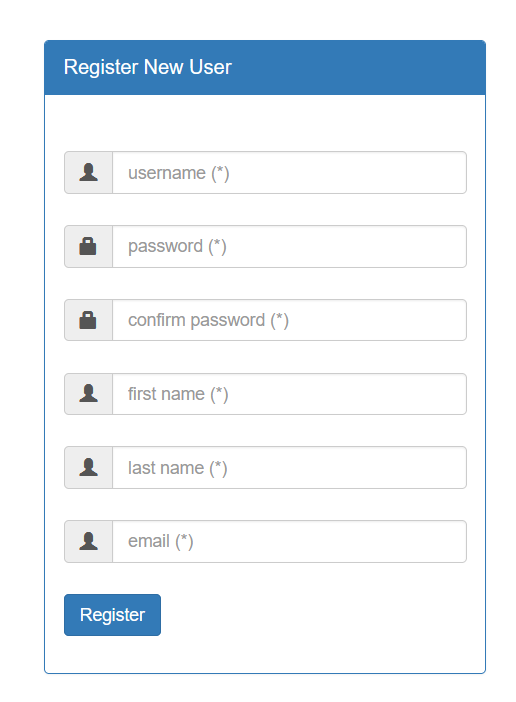
6.Ekrany aplikacji:

=================================================================================



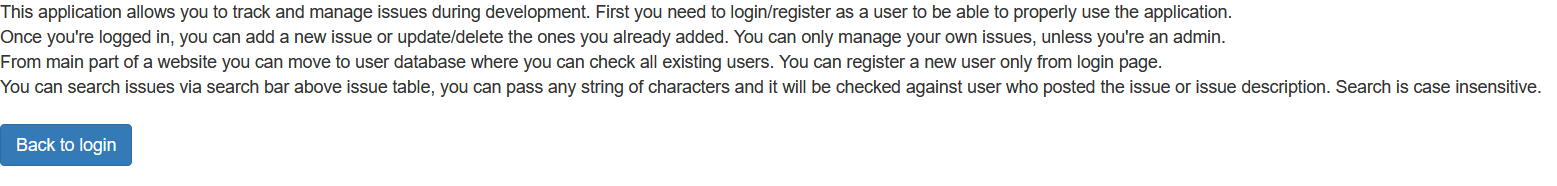
* Logowanie: możliwość zalogowania się na istniejącego użytkownika, zarejestrowania nowego lub przeczytania instrukcji.

=================================================================================



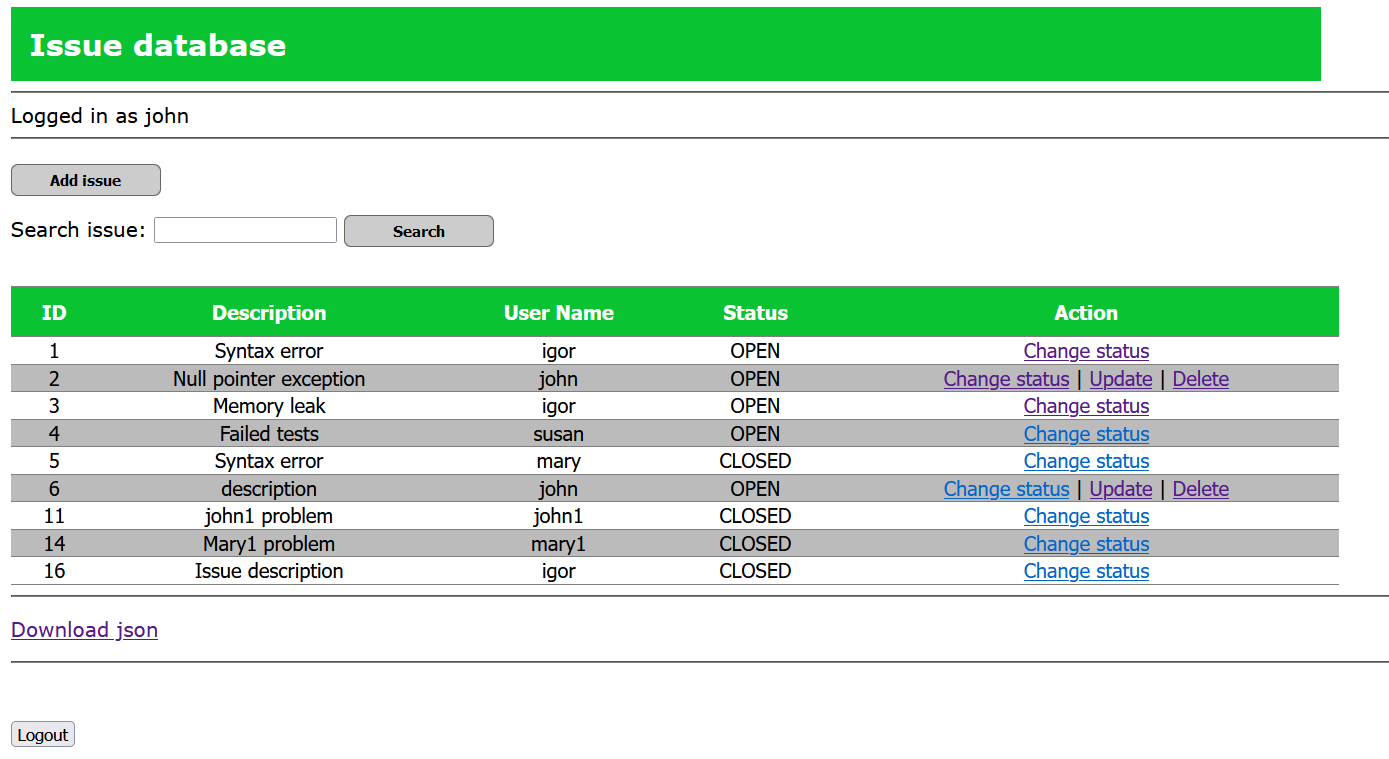
* Rejestracja nowego użytkownika: po wypełnieniu wszystkich pól zostanie utworzony nowy użytkownik.
* Pole „username” musi być unikalne, nie można utworzyć dwóch użytkowników o tej samej nazwie.
* Spring security framework odpowiada za zabezpieczanie logowania/rejestracji.

=================================================================================



* Instrukcja: krótka instrukcja mówiąca o obsłudze aplikacji.

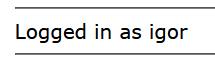
=================================================================================



=================================================================================

Główny ekran aplikacji:

=================================================================================



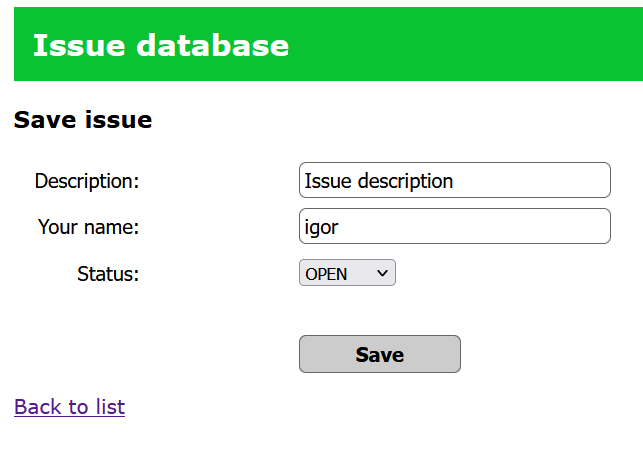
* Username aktualnie zalogowanego użytkownika.

=================================================================================



* Przycisk dodania nowego wpisu

=================================================================================



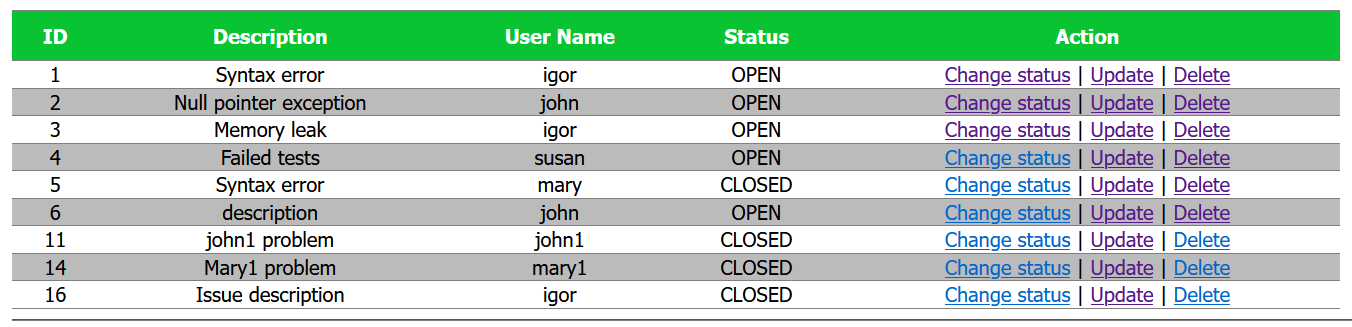
* Ekran dodania nowego wpisu.
* Pole „Your name:” musi zgadzać się z nazwą użytkownika dodającego.

=================================================================================



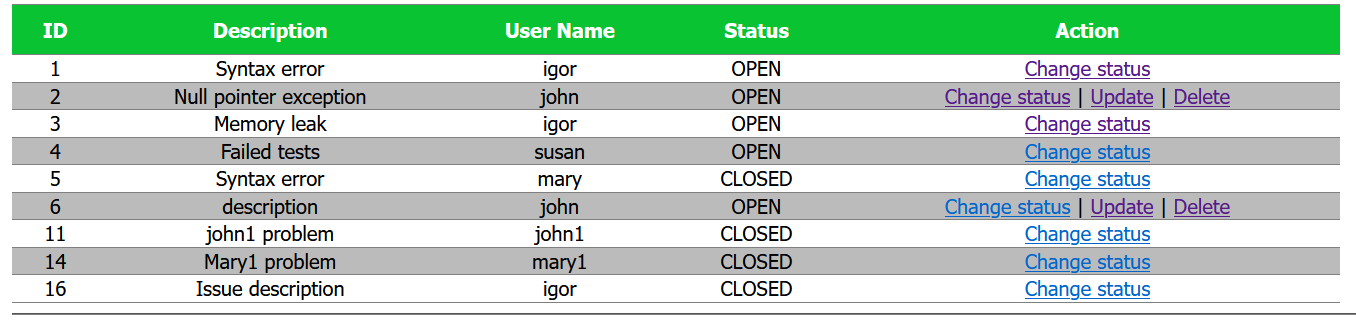
* Pole wyszukiwania: pozwala wyszukać wpis podając podciąg opisu lub nazwy użytkownika, który wpis dodał.
* Wielkość liter nie ma znaczenia.

=================================================================================



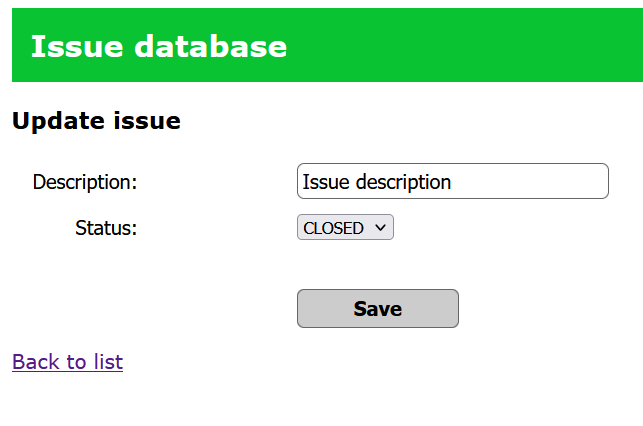
* Główna tabela bazy, pokazuje wszystkie wpisy, ich dane oraz akcje jakie można podjąć.

=================================================================================



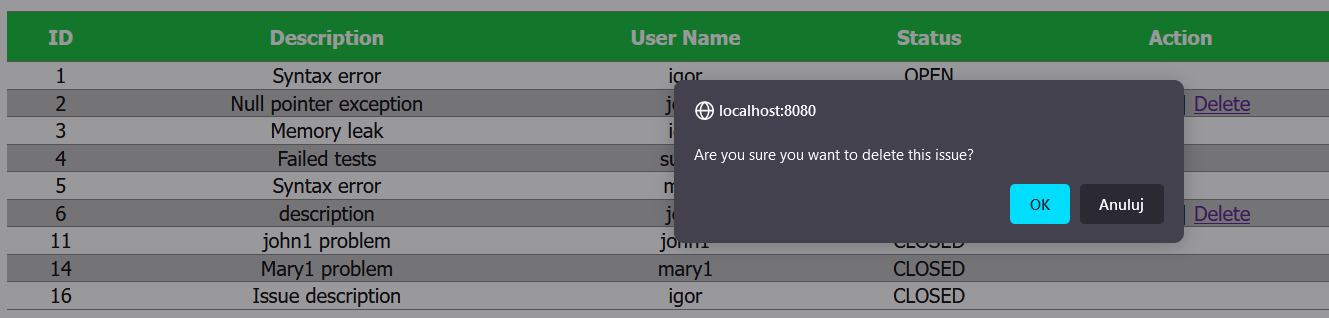
* Po zalogowaniu się na użytkownika o nazwie „john” możemy edytować tylko wpisy przez niego zamieszczone.

=================================================================================



* Ekran edycji wpisu.

=================================================================================



* Popup przy usuwaniu wpisu.

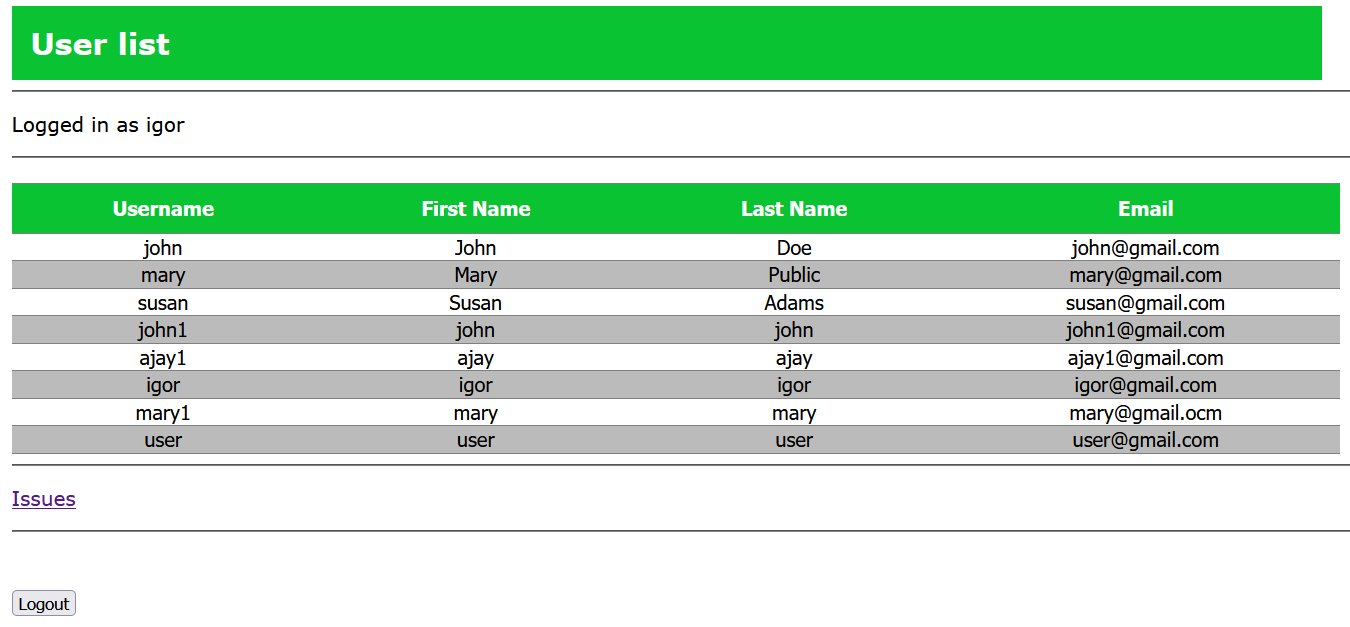
=================================================================================



=================================================================================

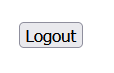
Hiperlinki do poruszania się po stronie:

* User list przeniesie nas do tabeli z danymi użytkowników. Dostępne tylko dla admina
* Download json: pobiera plik z wpisamy bazy jako json.



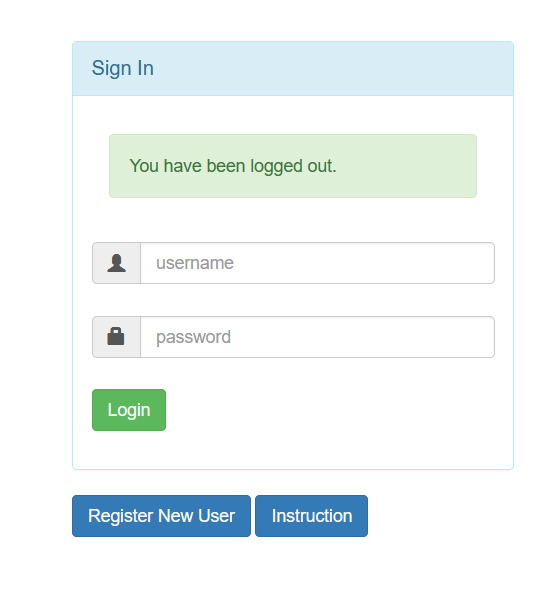
* Podobna struktura, widać dane wszystkich użytkowników, na dole link do poprzedniej strony i wylogowania.

=================================================================================



* Przycisk do wylogowania.

=================================================================================



* Ekran po wylogowaniu.

=================================================================================