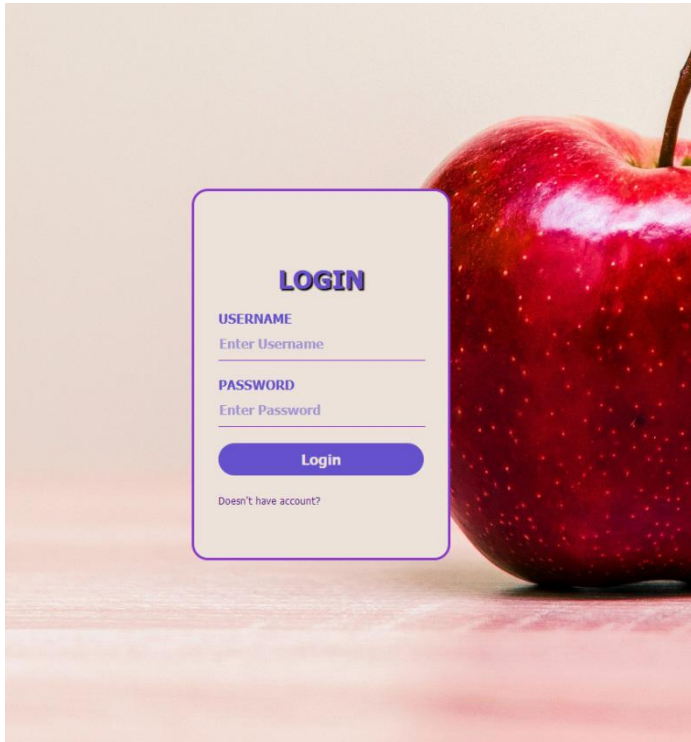


Projekt TS

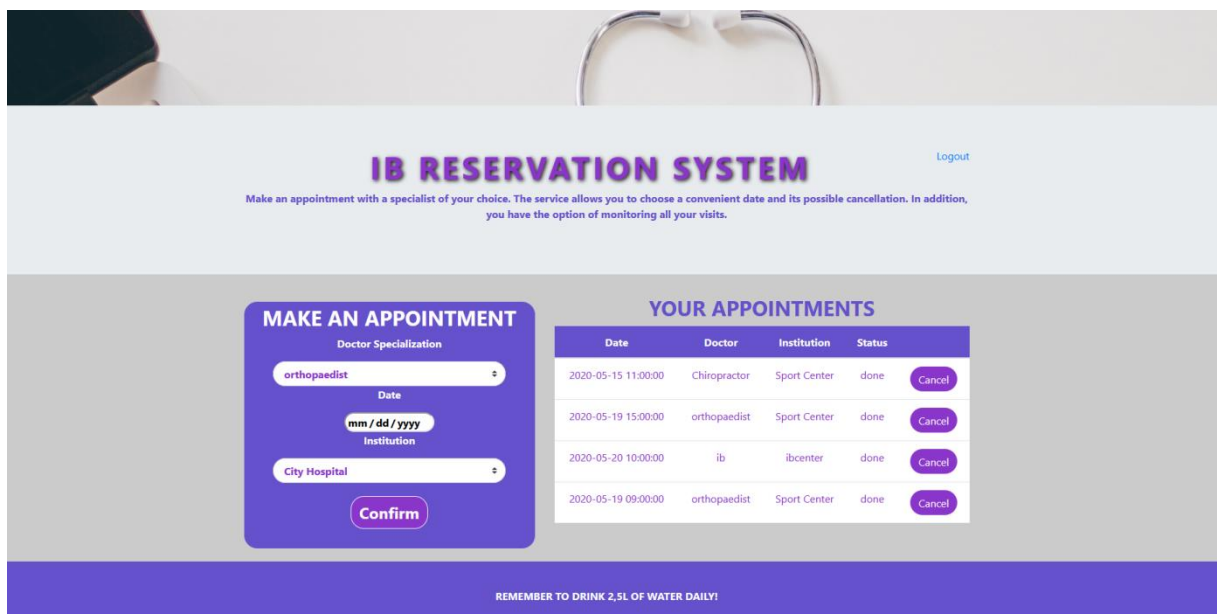
Michał Marciniak 244811

Repozytorium: <https://github.com/MMarciniak103/ReservationService>

Najpierw trafiamy na ekran logowania:



Po poprawnym zalogowaniu trafiamy do głównego ekranu aplikacji:



Mamy możliwość umówić się na wizytę, a w tabelce po prawej są pokazane wszystkie nasze wizyty. Istnieje możliwość ich anulowania poprzez przycisk "Cancel".

Choose Time

7:00 7:00 8:00 9:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00 19:00 20:00

MAKE AN APPOINTMENT

Doctor Specialization: orthopaedist

Date: 05 / 19 / 2020

Institution: Sport Center

Confirm

YOUR APPOINTMENTS

Date	Doctor	Institution	Status
2020-05-15 11:00:00	Chiropractor	Sport Center	done
2020-05-19 15:00:00	orthopaedist	Sport Center	done
2020-05-20 10:00:00	ib	ibcenter	done
2020-05-19 09:00:00	orthopaedist	Sport Center	done

REMEMBER TO DRINK 2.5L OF WATER DAILY!

Użytkownik jest poinformowany o dostępnych terminach za pomocą kolorów.

DOCTOR DOESNT WORK IN THIS HOURS!

MAKE AN APPOINTMENT

Doctor Specialization: orthopaedist

Date: 05 / 15 / 2020

Institution:

Date

2020-05-15

2020-05-19

2020-05-20

Aplikacja informuje użytkownika o powodzeniu jego akcji, poprzez wyświetlanie różnych komunikatów. Powyżej przykładowa wiadomość, mówiąca, że doktor nie pracuje w danych godzinach.

IB RESERVATION SYSTEM

Make an appointment with a specialist of your choice. The service allows you to choose a convenient date and its possible cancellation. In addition, you have the option of monitoring all your visits.

Admin Panel

Logout

MAKE AN APPOINTMENT

Doctor Specialization

orthopaedist

Date

mm / dd / yyyy

Institution

City Hospital

Confirm

YOUR APPOINTMENTS

Date	Doctor	Institution	Status
2020-05-13 10:00:00	dentist	Dental Center	done Cancel
2020-05-18 12:00:00	orthopaedist	Sport Center	done Cancel
2020-05-10 14:00:00	orthopaedist	Sport Center	done Cancel

Będąc zalogowanym na koncie admina, mamy możliwość przejść do panelu umożliwiającego dodawanie lekarzy do bazy danych.

ADD DOCTOR

SPECIALIZATION

testowy

INSTITUTION

tester

WORKING FROM:

7

WORKING TO:

15

Add

INSTITUTION: TESTER

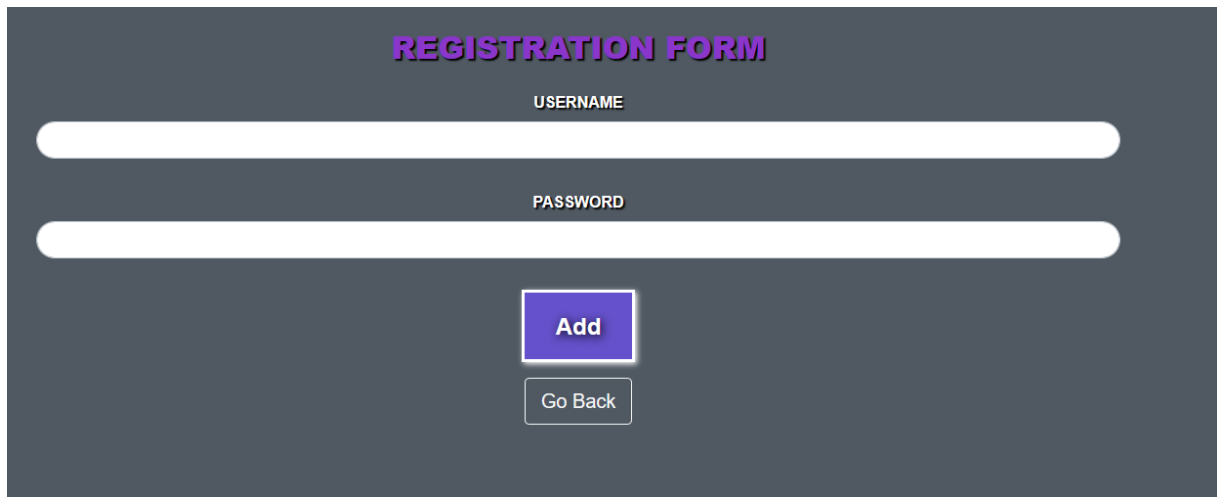
SPECIALIZATION: TESTOWY

WORKING FROM: 7 TO 15

Go Back

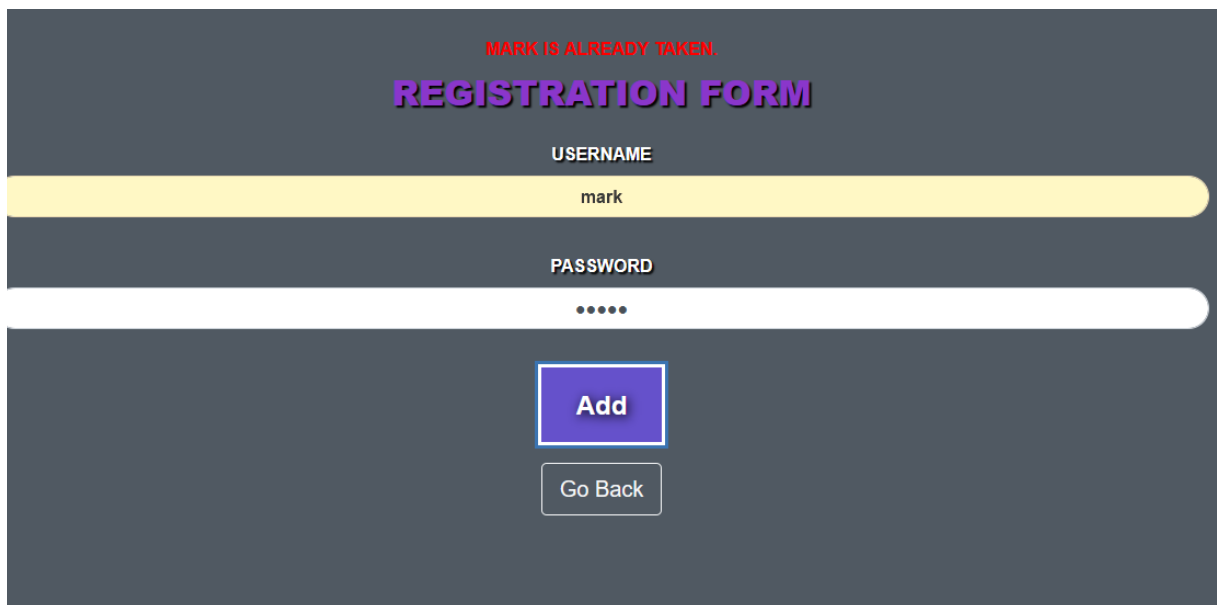
Przykład panelu admina.

Cofając się do ekranu logowania, mamy możliwość zarejestrowania się w bazie danych użytkowników.



A screenshot of a registration form titled "REGISTRATION FORM" in purple. It features two white input fields for "USERNAME" and "PASSWORD". Below the fields are two buttons: a purple "Add" button and a grey "Go Back" button.

Użytkownik nie może wybrać nazwy, która jest już zajęta.



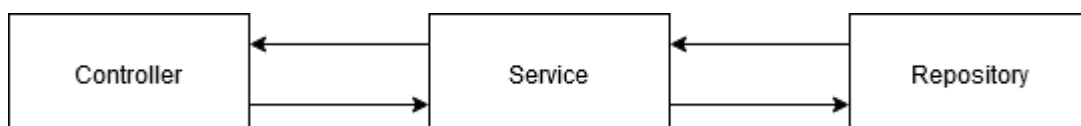
A screenshot of the registration form showing an error state. At the top, a red message "MARK IS ALREADY TAKEN." is displayed. The "REGISTRATION FORM" title is in purple. The "USERNAME" field, containing the text "mark", is highlighted with a yellow background. The "PASSWORD" field is empty and masked with dots. The "Add" and "Go Back" buttons are at the bottom.

Aplikacja jest napisana poprzez stosowanie wzorca MVC oraz DTO. Komunikacja pomiędzy klientem a serwerem odbywa się poprzez zapytania http.

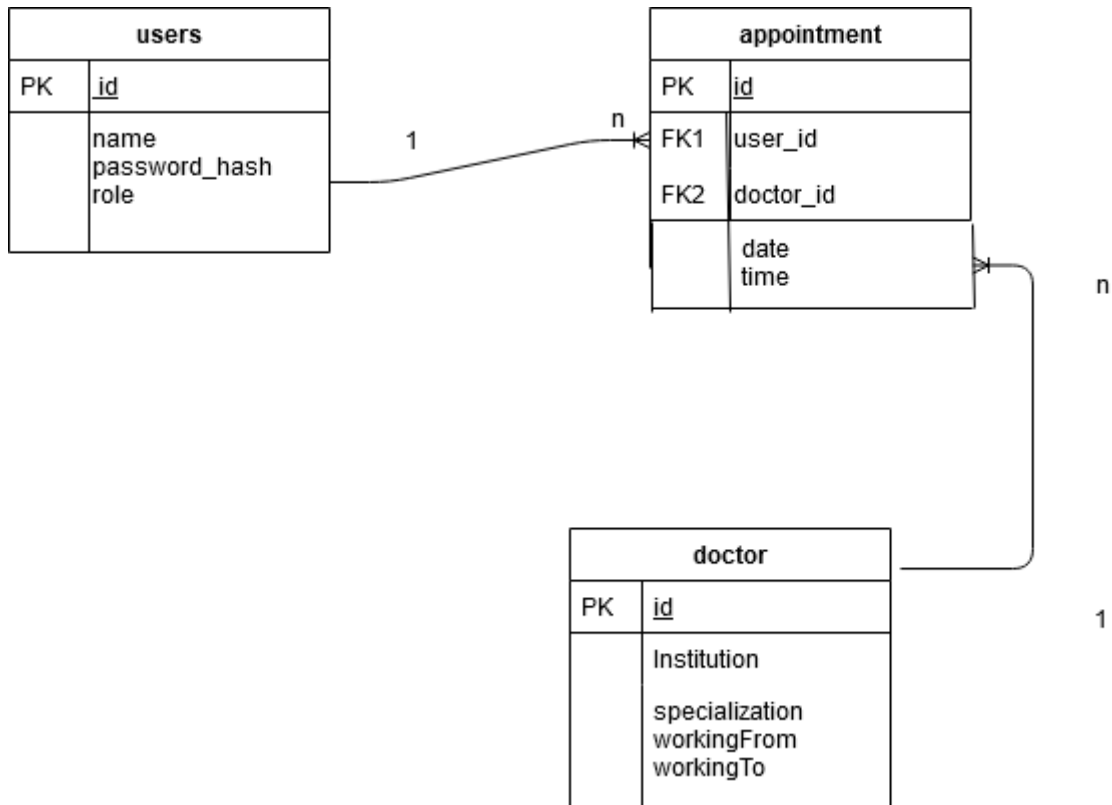
Warstwa obsługująca zapytania klienta i widoki nosi nazwę kontrolera.

Kontroler komunikuje się z warstwą serwisu, która implementuje logikę biznesową.

Najniższą warstwą, komunikującą się z bazą danych jest repozytorium.



Widoki jsp poprzez javascript wysyłają zapytania http do warstwy kontrolera. Wykorzystują do tego modele danych, które w momencie gdy mają być zapisane w bazie danych, zostają zamienione na klasy DTO przy wykorzystaniu DTOBuildera.



Encja appointment i user połączona jest relacją one-many (Wizyta odnosi się do konkretnego użytkownika , ale użytkownik może mieć wiele wizyt).

Analogicznie jest z relacją doctor - appointment.

Dodatkowo w bazie danych ustawiony jest scheduler, który co godzinę wywołuje procedurę aktualizującą stan wizyt. Sprawdza ona czy dana wizyta jest już po terminie, i jeśli jest to zmienia jej status na 'done'.