

Natalia Jaśkowska:
natalia.jaskowska.stud@pw.edu.pl
313900

Warszawa, 28.10.2022

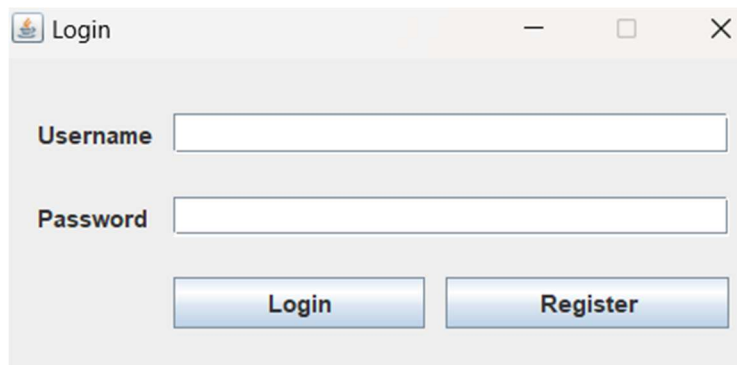
Grzegorz Molak:
grzegorz.molak.stud@pw.edu.pl
310280
OPA – Java - obiektowe programowanie aplikacji
Semestr 22Z

System umawiania wizyty u lekarza w JAVIE

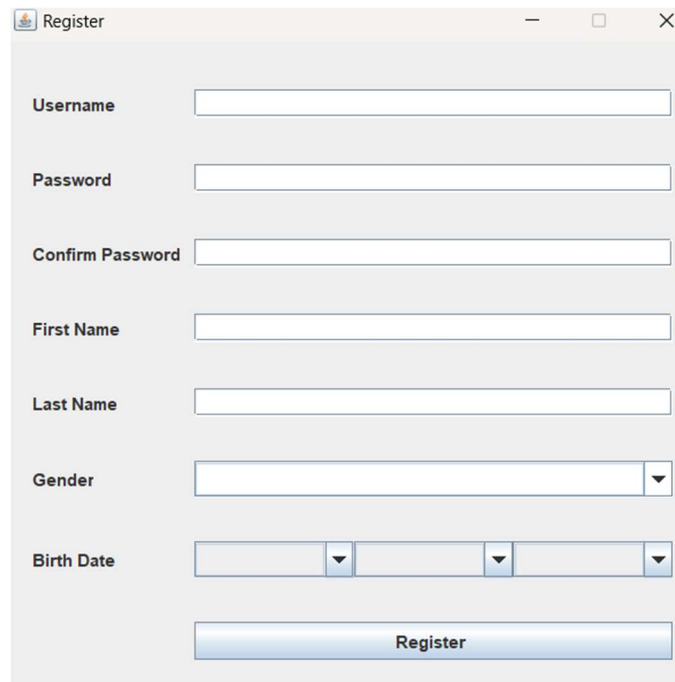
Celem projektu było zrealizowanie systemu umożliwiającego użytkownikowi rejestrację na wizytę u lekarza z wykorzystaniem Javy. Aplikacja połączona jest z bazą danych MariaDB za pomocą JDBC oraz udostępnia użytkownikowi interfejs graficzny wykonany przy użyciu biblioteki SWING.

Działanie programu

Po uruchomieniu programu użytkownik zostaje poproszony o zalogowanie do aplikacji za pomocą swojego login oraz hasła.

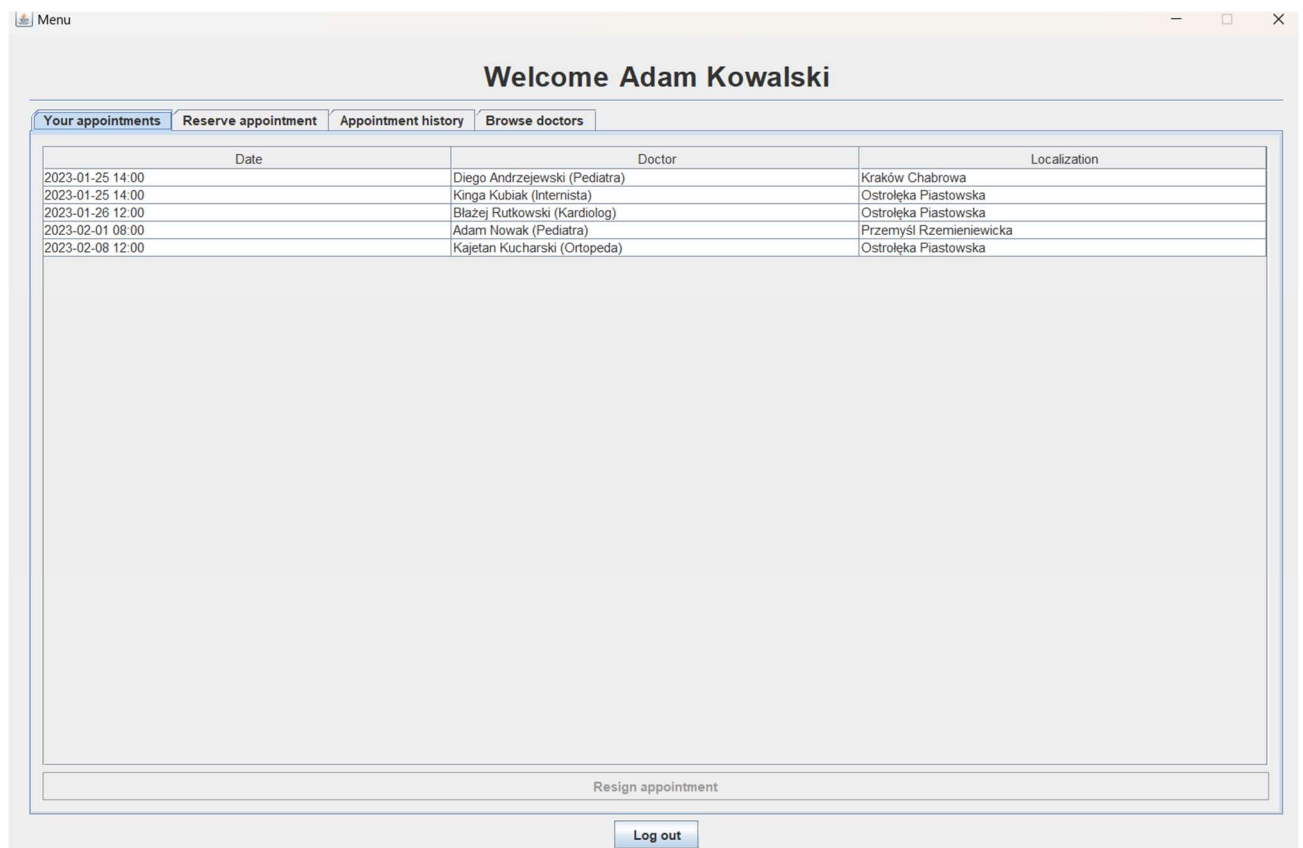
A screenshot of a Java Swing window titled "Login". The window has a standard title bar with minimize, maximize, and close buttons. Inside the window, there are two text input fields. The first field is labeled "Username" and the second is labeled "Password". Below these fields are two buttons: "Login" and "Register". The buttons have a blue gradient and a 3D effect.

W przypadku gdy użytkownik nie posiada konta, może się zarejestrować, uzupełniając pola odpowiednimi wartościami i zatwierdzając.



A screenshot of a web application window titled "Register". The window contains several input fields for user registration: Username, Password, Confirm Password, First Name, Last Name, Gender (a dropdown menu), and Birth Date (three separate dropdown menus for day, month, and year). At the bottom of the form is a blue "Register" button.

Po zalogowaniu użytkownikowi ukazuje się lista jego nadchodzących wizyt, z możliwością natychmiastowej rezygnacji. Dostępne są również przyciski umożliwiające nawigację, a przy tym dostęp do innych funkcji systemu takich jak: rezerwacja wizyty, przegląd przeszłych wizyt oraz przegląd lekarzy wraz z ich tygodniowymi zmianami.



A screenshot of a web application window titled "Menu". The main heading is "Welcome Adam Kowalski". Below the heading are four tabs: "Your appointments", "Reserve appointment", "Appointment history", and "Browse doctors". The "Your appointments" tab is active, displaying a table of upcoming appointments. The table has three columns: Date, Doctor, and Localization. Below the table is a "Resign appointment" button. At the bottom of the window is a "Log out" button.

Date	Doctor	Localization
2023-01-25 14:00	Diego Andrzejewski (Pediatria)	Kraków Chabrowa
2023-01-25 14:00	Kinga Kubiak (Internista)	Ostrolęka Piastowska
2023-01-26 12:00	Błażej Rutkowski (Kardiolog)	Ostrolęka Piastowska
2023-02-01 08:00	Adam Nowak (Pediatria)	Przemyśl Rzemieniewicka
2023-02-08 12:00	Kajetan Kucharski (Ortopeda)	Ostrolęka Piastowska

Wyszukanie pasującej użytkownikowi wizyty ułatwione jest poprzez filtrację wyników w takich kategoriach jak: specjalizacja lekarza, ograniczenie do konkretnego lekarza, lokalizacja oraz przedział czasowy, w którym ma się znajdować wizyta.

Menu

Welcome Adam Kowalski

Your appointments | Reserve appointment | Appointment history | Browse doctors

Date	Doctor	Localization
2023-01-25 12:00	Adela Ziolkowska (Dermatolog)	Opole Uroczna
2023-01-25 13:00	Adela Ziolkowska (Dermatolog)	Opole Uroczna
2023-01-25 14:00	Adela Ziolkowska (Dermatolog)	Opole Uroczna
2023-01-25 15:00	Adela Ziolkowska (Dermatolog)	Opole Uroczna
2023-01-26 08:00	Adela Ziolkowska (Dermatolog)	Opole Uroczna
2023-01-26 09:00	Adela Ziolkowska (Dermatolog)	Opole Uroczna
2023-01-26 10:00	Adela Ziolkowska (Dermatolog)	Opole Uroczna
2023-01-26 11:00	Adela Ziolkowska (Dermatolog)	Opole Uroczna
2023-01-26 12:00	Adela Ziolkowska (Dermatolog)	Opole Uroczna
2023-01-26 13:00	Adela Ziolkowska (Dermatolog)	Opole Uroczna
2023-01-26 14:00	Adela Ziolkowska (Dermatolog)	Opole Uroczna
2023-01-26 15:00	Adela Ziolkowska (Dermatolog)	Opole Uroczna

Specialization: Dermatolog

Doctor: Adela Ziolkowska (Dermatolog)

Location: Opole Uroczna

First date: 2023 JANUARY 25

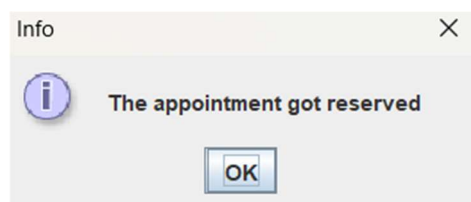
Last date: 2023 JANUARY 27

Search

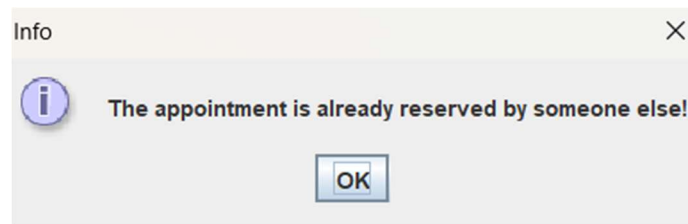
Reserve appointment

Log out

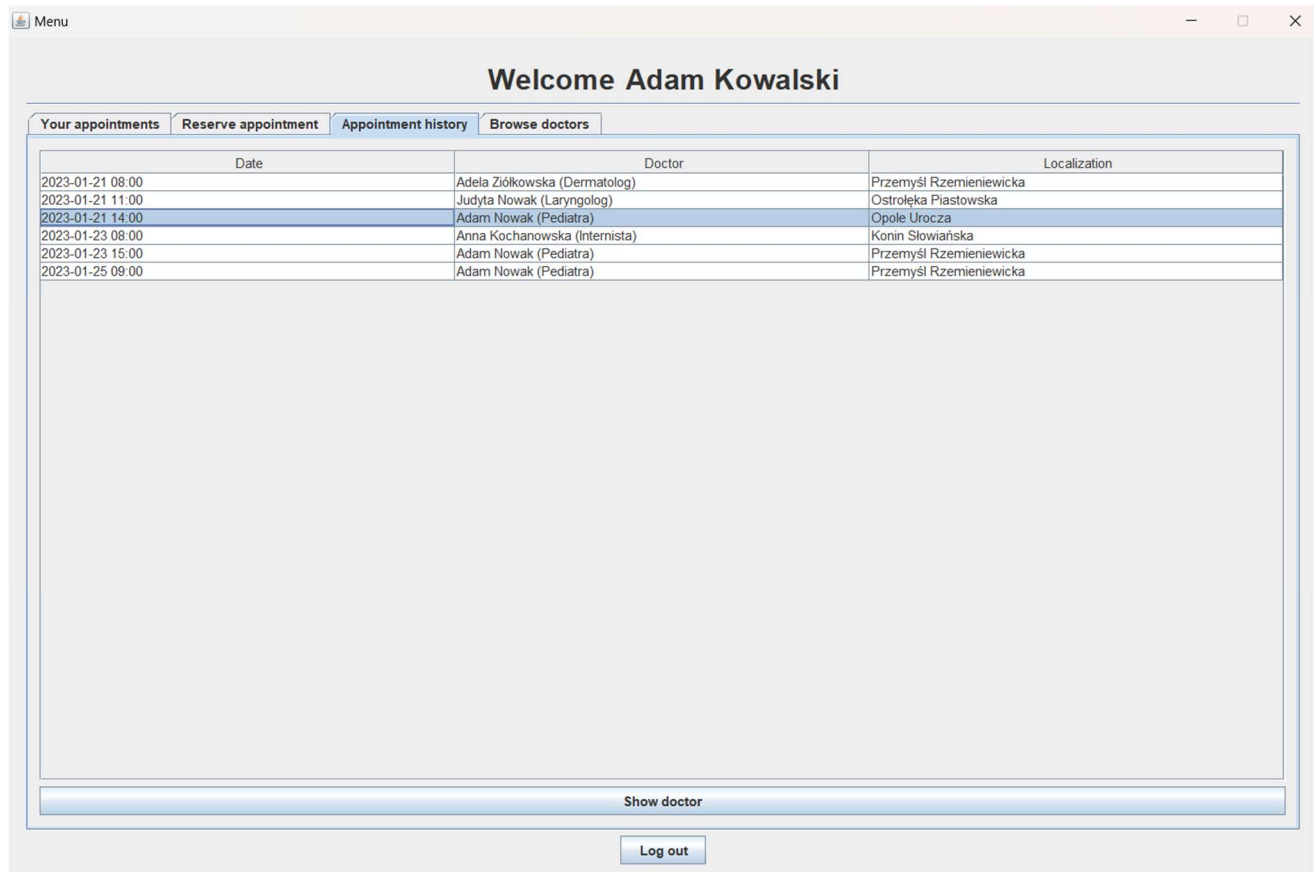
Wybór wizyty i kliknięcie *Reserve appointment* powoduje rezerwację wybranej wizyty i przejście z powrotem do panelu nadchodzących wizyt



W przypadku gdy po pokazaniu możliwości użytkownikowi, ktoś inny zarezerwował wizyt, nie powiedzie się to, a użytkownik otrzyma stosowny komunikat



Historia wizyt pozwala przeglądać odbyte wizyty oraz przeniesienie do panelu lekarza przeprowadzającego daną wizytę, przykładowo w celu sprawdzenia zmian lekarza w celu ponownego umówienia się na wizytę



Menu

Welcome Adam Kowalski

[Your appointments](#)
[Reserve appointment](#)
[Appointment history](#)
[Browse doctors](#)

Adam Nowak

First name: Adam
 Last name: Nowak
 Gender: m
 Specialization: Pediatria

Localization	Day	Start time	End time
Przemyśl Rzemieniewicka	MONDAY	08:00	16:00
Przemyśl Rzemieniewicka	TUESDAY	08:00	16:00
Przemyśl Rzemieniewicka	WEDNESDAY	08:00	16:00
Rybnik Słowińska	THURSDAY	08:00	16:00
Kraków Chabrowa	FRIDAY	08:00	16:00
Opole Urocza	SATURDAY	08:00	16:00

Shifts

Log out

Struktura aplikacji

Aplikacji składa się z 3 klas służących do uruchomienia i testowania aplikacji:

Main – do uruchomienia aplikacji

Test – do testowania zdolności aplikacji do połączeń z bazą danych

Initialize – do zainicjowania i wstępnego wypełnienia prostej bazy danych do obsługi aplikacji

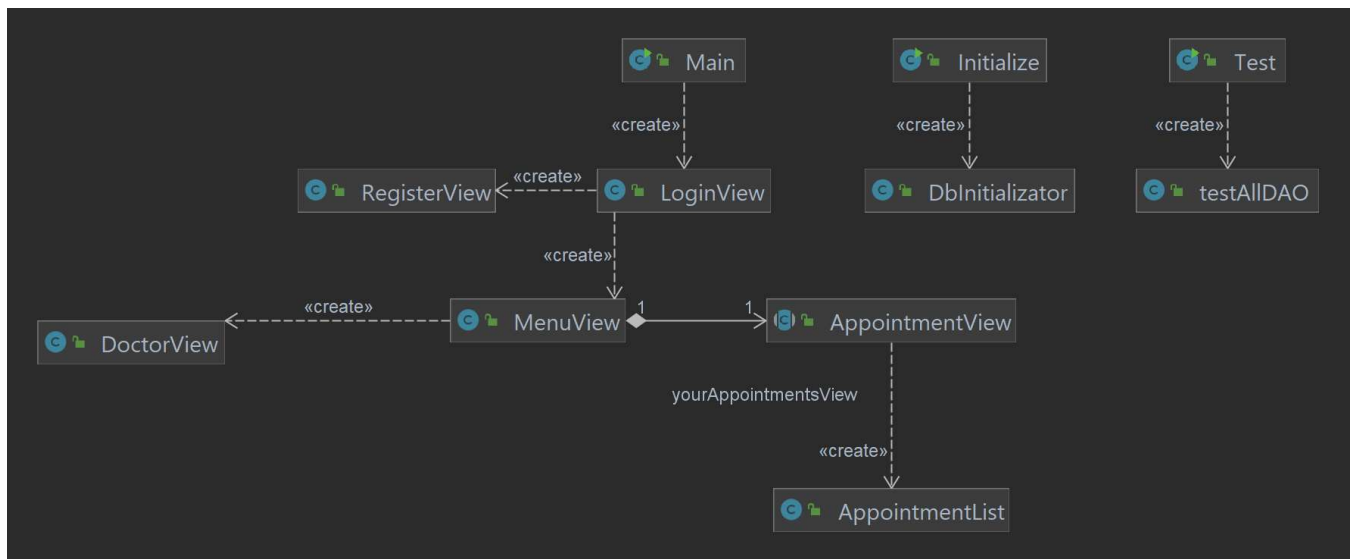
Reszta plików została podzielona na 3 paczki

Model – znajdują się tu klasy obiektów używanych do wyświetlania danych użytkownikowi, takie jak: Patient, Doctor, Appointment

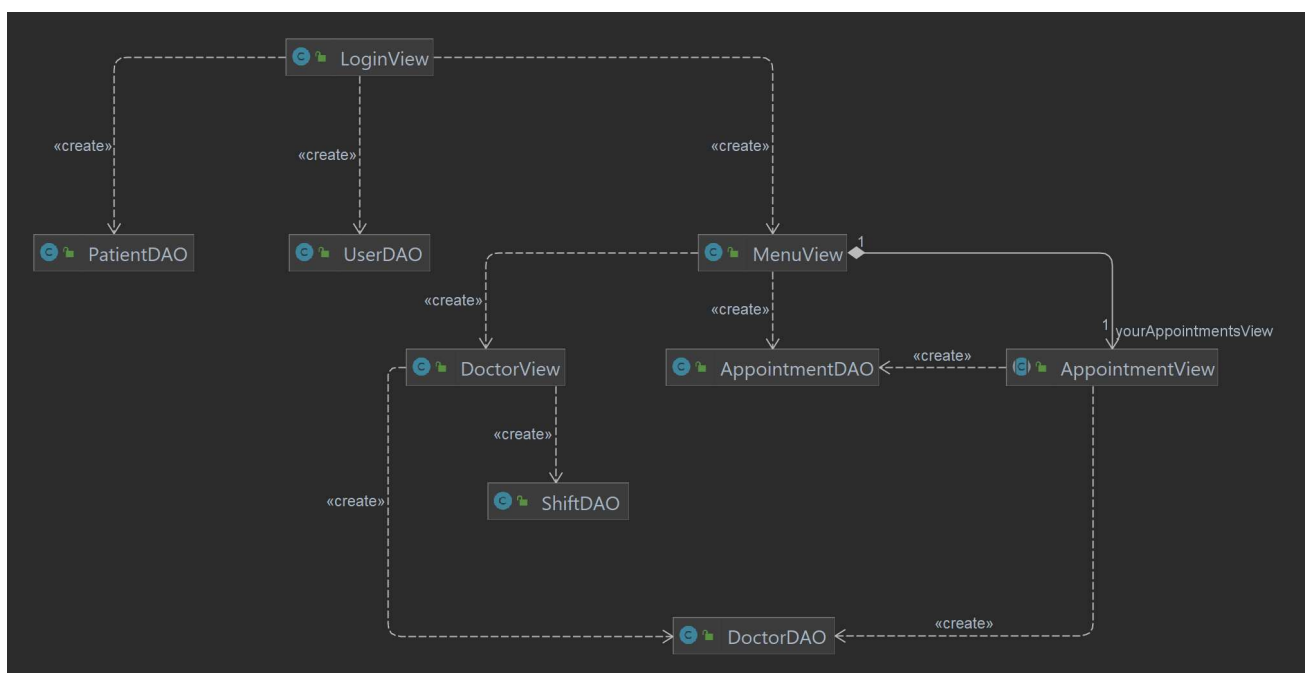
DAO – znajdują się tu klasy służące do obsługi bazy danych, takie jak połączenia z nią oraz pobieranie czy zapis obiektów klas należących do paczki Model z/do bazy danych.

Vlew – odpowiada za interfejs graficzny i obsługę zdarzeń z nim związanych.

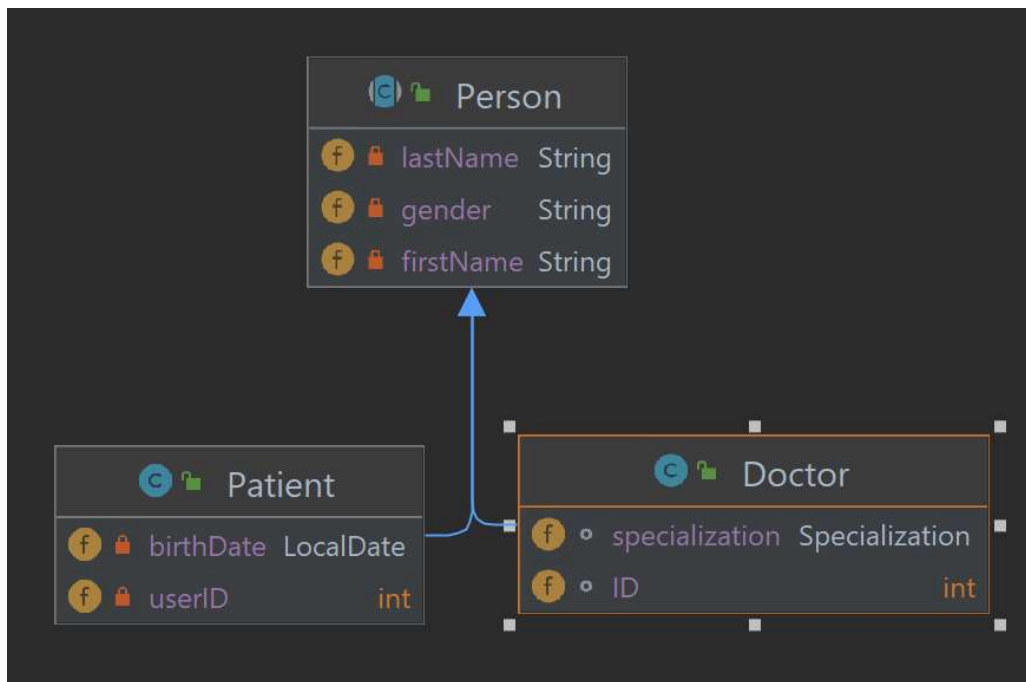
Schematy niektórych klas i zależności między nimi



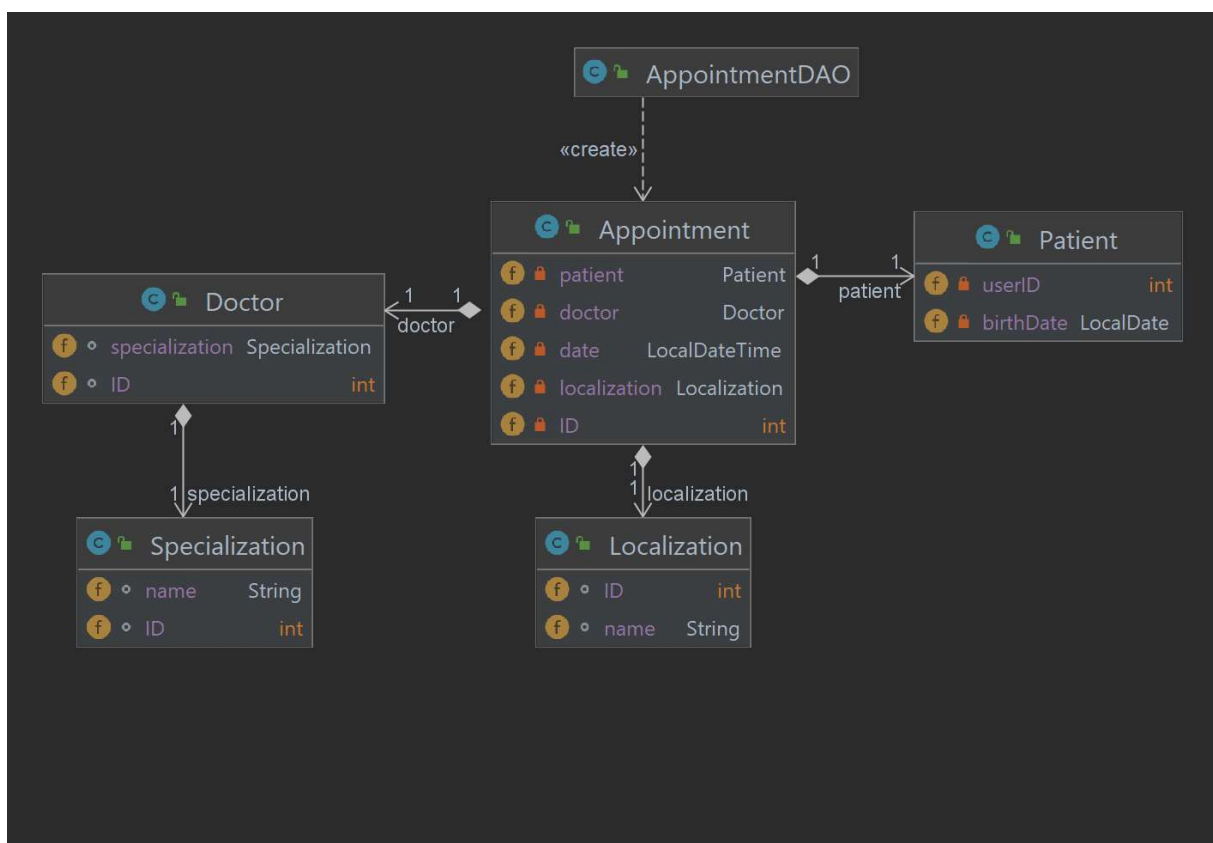
Powyższy schemat przedstawia w uproszczony sposób działanie programu. Klasa Main tworzy obiekt klasy odpowiadającej za logowanie, która z kolei umożliwia rejestrację bądź przejście do głównego widoku aplikacji. Główny widok aplikacji umożliwia przemieszczanie między niektórymi widokami



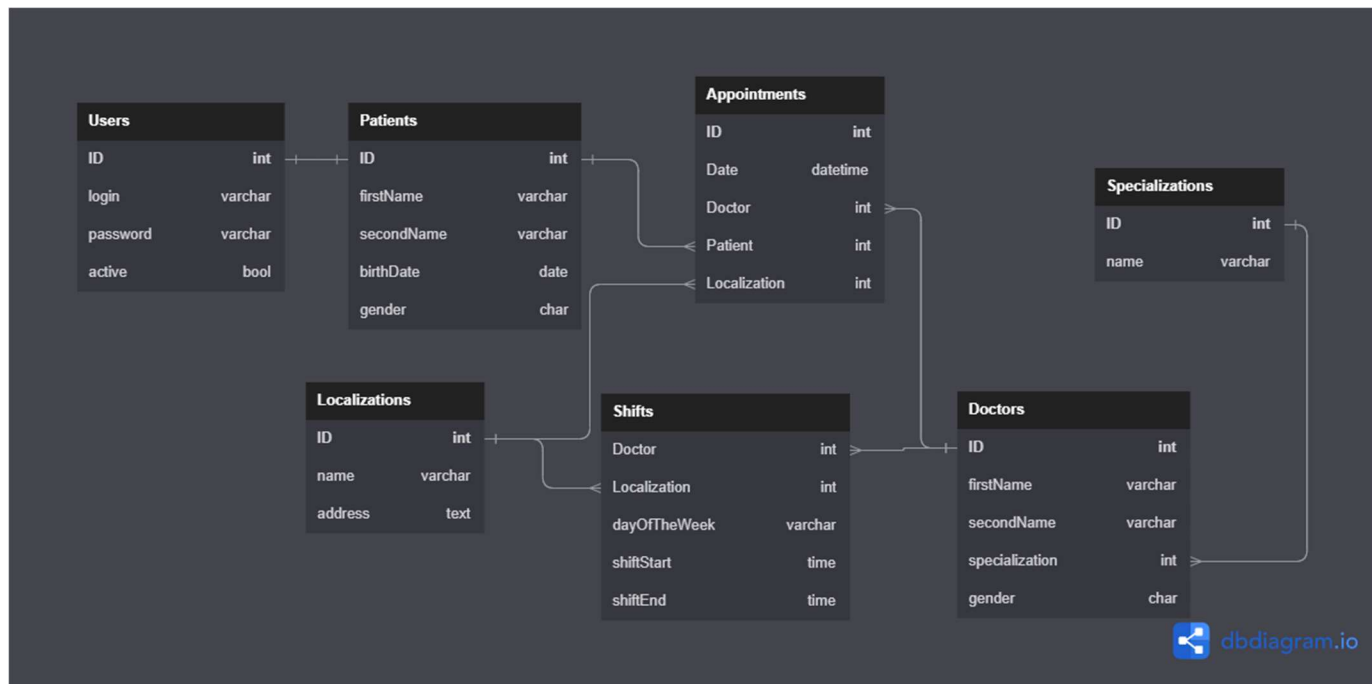
Powyższy schemat przedstawia to, z których bazodanowych klas korzystają poszczególne elementy widoku.



Klasy Patient i Doctor dziedziczą po abstrakcyjnej klasie Person



Powyższy schemat przedstawia tworzenie najbardziej złożonej klasy występującej w modelu aplikacji – wizyty. Wizyta posiada przypisanego pacjenta(lub null – jeżeli jest wolna), doktora, datę i godzinę rozpoczęcia, lokalizację oraz unikalne ID.



Schemat pokazujący tabele w bazach danych, ich pola oraz relacje między nimi.