

# Wprowadzenie

Celem aplikacji projektowej jest umożliwienie lekarzom sprawnego zarządzania pacjentami, których mają pod opieką oraz wizytami, które planują. Aplikacja umożliwia dodawanie, edytowanie oraz usuwanie pacjentów oraz wizyt z bazy danych.

# Strona techniczna

Projekt został wykonany w języku programowania Java w środowiskach IntelliJ i VisualStudioCode. Do stworzenia interfejsu graficznego wykorzystana została biblioteka JavaFX oraz program SceneBuilder jego zaprojektowania. Aplikacja zawiera 21 klas oraz 11 plików fxml odpowiadających oknom interfejsu. Stworzone klasy to:

- App – główna klasa odpowiadająca za uruchamianie aplikacji i nieużywany interfejs konsolowy

- Appointment – klasa obsługująca obiekty wizyty

- Person – klasa będąca bazą dla klas odnoszących się do ludzi

- Patient – klasa dziedzicząca po Person, obsługuje obiekty pacjentów

- Doctor - klasa dziedzicząca po Person, obsługuje obiekty lekarzy

- Klasy …Controller – klasa obsługująca dane okienko interfejsu graficznego

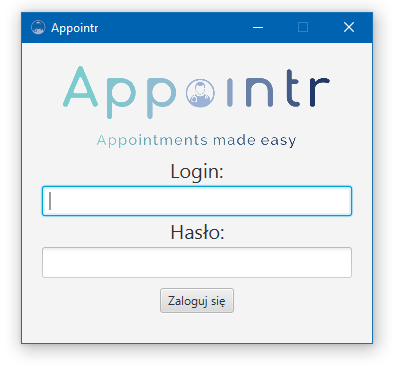
- Populate – klasa służąca do tworzenia testowej grupy ludzi do używania w programie

- AppointmentTableRow – klasa służąca do wypełniania tabeli wizyt w interfejsie graficznym

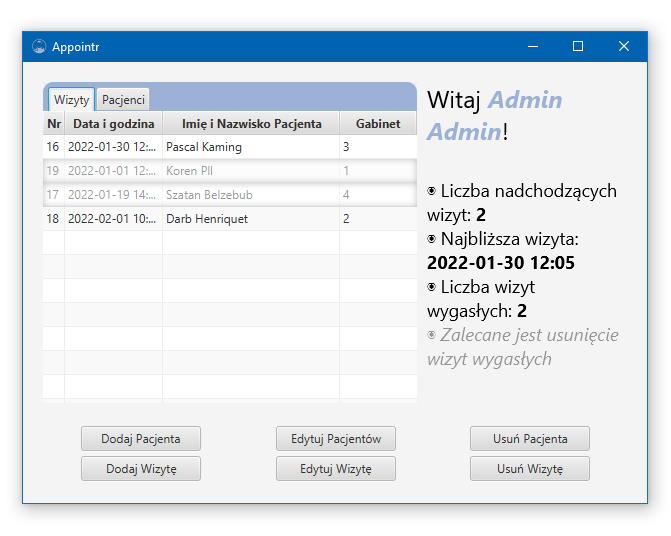
- PatientTableRow – klasa służąca do wypełniania tabeli wizyt w interfejsie graficznym

# Instrukcja obsługi

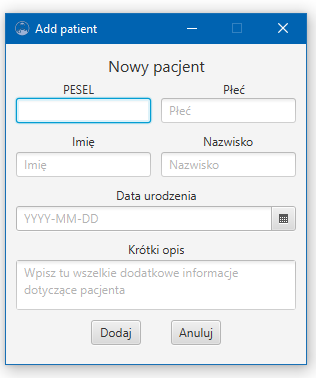
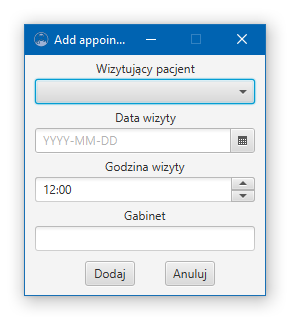
Aby rozpocząć korzystanie z aplikacji należy zalogować się używając loginu i hasła przypisanego do danego lekarza. Login i hasło należy wprowadzić do odpowiadających im pól, a następnie nacisnąć przycisk „Zaloguj się”.



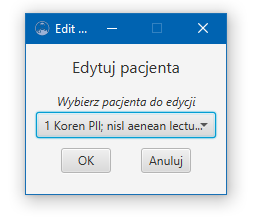
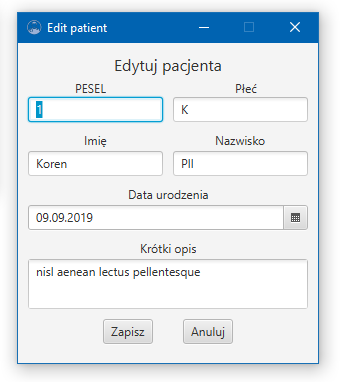
Po zalogowaniu się do dyspozycyjny staje się panel główny zawierający główne funkcjonalności oprogramowania. Tabela wyświetla wizyty, które są przypisane do zalogowanego doktora. Te z wizyt, które są w kolorze jasno szarym uznawane są za wygasłe, gdyż ich data i godzina są przed obecną. W zakładce „Pacjenci” znajdują się kluczowe informacje dotyczące pacjentów wpisanych do bazy danych. Klikanie na nagłówki kolumn pozwala na sortowanie wierszy po zawartości danych kolumn. Używanie przycisków pozwala na korzystanie z wymienionych w nich opcji.

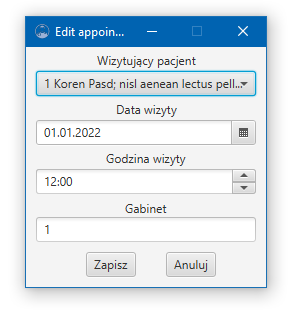
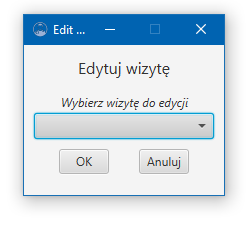


Aby dodać pacjenta należy kliknąć przycisk „Dodaj pacjenta”, wypełnić pojawiający się kwestionariusz oraz zatwierdzić przyciskiem „Dodaj”. Akcję tą można anulować odpowiednim przyciskiem. W analogiczny sposób można postąpić przy chęci dodania wizyty.

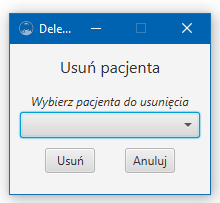
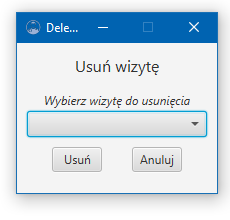


W celu edycji pacjentów i wizyt należy najpierw wybrać pacjenta, który będzie edytowany z listy, nacisnąć przycisk OK., a następnie przystąpić do edycji w pojawiającym się kolejnym oknie. Analogicznie należy postąpić w przypadku edycji wizyt.





Usuwanie wizyt odbywa się przez wybór pacjenta lub wizyty z listy i naciśnięcie przycisku usuń.



# Testowanie

Testowanie odbywało się poprzez wykonywanie czynności w programie i porównywaniu ich z efektami, które były oczekiwane w bazie danych oraz poprzez kontrolę zachowania interfejsu graficznego. Kod był również na bieżąco sprawdzany przez wszystkich uczestników projektu, aby uniknąć potencjalnych błędów i niedociągnięć.

# Zasługi indywidualne

Bartłomiej Dudek:

- Stworzenie repozytorium

- Stworzenie koncepcji interfejsu graficznego

- Utworzenie klas Doctor, Patient, Appointment

- Dodanie do projektu bibliotek do obsługi plików fxml

- Stworzenie konsolowego interfejsu

- Stworzenie elementów interfejsu graficznego służących do dodawania, edytowania, usuwania wizyt i pacjentów oraz ekranu głównego

- Dodanie obsługi wyżej wymienionych elementów do aplikacji

- Ustawienie przekazywania danych pomiędzy wszystkimi elementami interfejsu graficznego

- Dodanie połączenia z bazą danych do niektórych elementów interfejsu graficznego

- Stworzenie dokumentacji