****

**Programowanie obiektowe**

*Program do zarządzania bazą danych szpitalnego oddziału ratunkowego „Szpitalex”*

Prowadzący: Autor:

mgr inż. Ewa Żesławska Tomasz Witas

w65518

Kierunek: 3 IIZ/2021, grupa GP02

Rzeszów, r.a. 2022/2023

Spis treści

[**1.** **Cele projektu** 3](#_Toc125467490)

[**2.** **Opis techniczny projektu** 3](#_Toc125467491)

[**3.** **Harmonogram realizacji projektu** 3](#_Toc125467492)

[**4.** **Prezentacja warstwy użytkowej projektu** 4](#_Toc125467493)

[**5.** **Testy jednostkowe** 9](#_Toc125467494)

[**6.** **Repozytorium, system kontroli wersji** 9](#_Toc125467495)

[**7.** **Podsumowanie** 9](#_Toc125467496)

[**8.** **Literatura** 9](#_Toc125467497)

1. **Cele projektu**

Celem projektu było stworzenie programu umożliwiającego sprawną i bezproblemową obsługę bazy danych w Szpitalnych oddziałach ratunkowych, zapewniając jednocześnie maksymalną prostotę działania i podatność na implementację nowych funkcji.

**Wymagania funkcjonalne i niefunkcjonalne**

**Wymagania funkcjonalne**

* Nawiązanie połączenia oraz praca na bazie danych.
* Panel logowania do programu.
* Wyświetlanie, dodawanie i usuwanie, informacji o pracujących lekarzach.
* Wyświetlanie, dodawanie i usuwanie, informacji o przyjętych pacjentach.
* Wyświetlanie, dodawanie i usuwanie, informacji o przyjęciu konkretnego pacjenta, diagnozie jego przypadłości oraz obsługującym go lekarzu.
* Wyświetlanie szczegółów diagnozy dla wybranego przez użytkownika przyjęcia.
* Walidacja danych uniemożliwiająca podanie błędnych danych do bazy dla wszystkich pól formularza.
* Monity informujące o obsługiwanym formacie danych w formularzu.
* Możliwość wylogowania oraz ponownego zalogowania użytkownika.

**Wymagania niefunkcjonalne**

* Bardzo prosty i intuicyjny interfejs zapewniający pracę bez doświadczenia.
* Styl okien programu jak i wszystkich jego elementów dostosowany automatycznie zależnie od wersji systemu operacyjnego użytkownika.
* Wydajność i szybkość działania.
* Wydajna, szybka i bezpieczna obsługa bazy danych.
* Program stworzony w języku Java.
* Program nawiązuje połączenie z bazą danych i używa rekordów w niej zapisanych.

1. **Opis techniczny projektu**

* Środowisko programistyczne Javy: Java Oracle OpenJDK 17, IntelliJ IDEA.
* Baza danych: MySql.
* Dodatkowe biblioteki: mysql-connector-j-8.0.31.
* Jednostka komputerowa obsługująca język Java wraz z odpowiednim środowiskiem programistycznym.

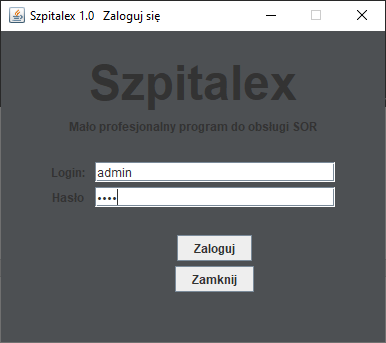
1. **Harmonogram realizacji projektu**

Poniżej zamieszono harmonogram realizacji projektu.

Rysunek 1. Diagram Gantta

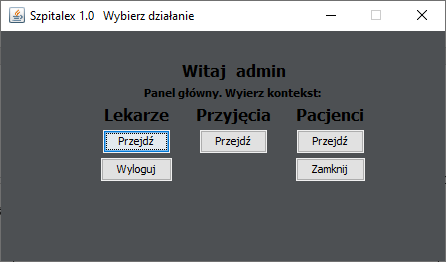
1. **Prezentacja warstwy użytkowej projektu**

Na rysunku 1 przedstawiono panel powitalny, będący jednocześnie panelem logowania do programu, które zostaje wyświetlony po uruchomieniu:



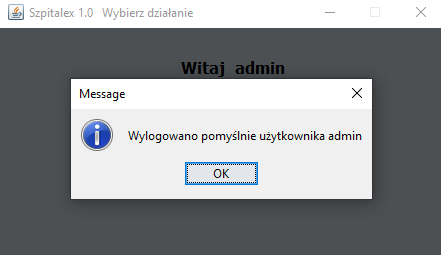
Rysunek 1. Panel logowania

Po poprawnym zalogowaniu otwiera się panel menu głównego z przyciskami umożliwiającymi dojście do wybranych działów (Rysunek 2).



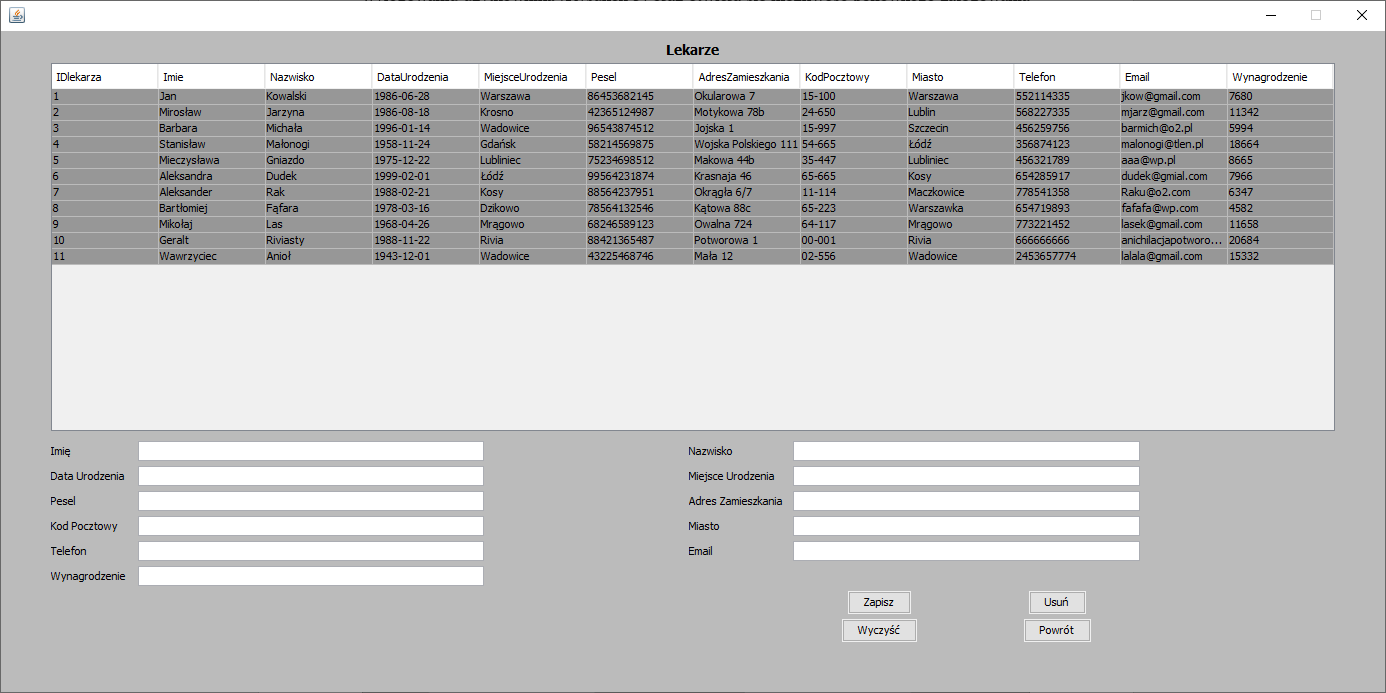
Rysunek 2. Menu główne

Z panelu menu po wciśnięciu przycisku „wyloguj” otrzymujemy monit o poprawnym wylogowaniu użytkownika (Rysunek 3) oraz pojawia się możliwość ponownego zalogowania na inne konto (Rysunek 1).

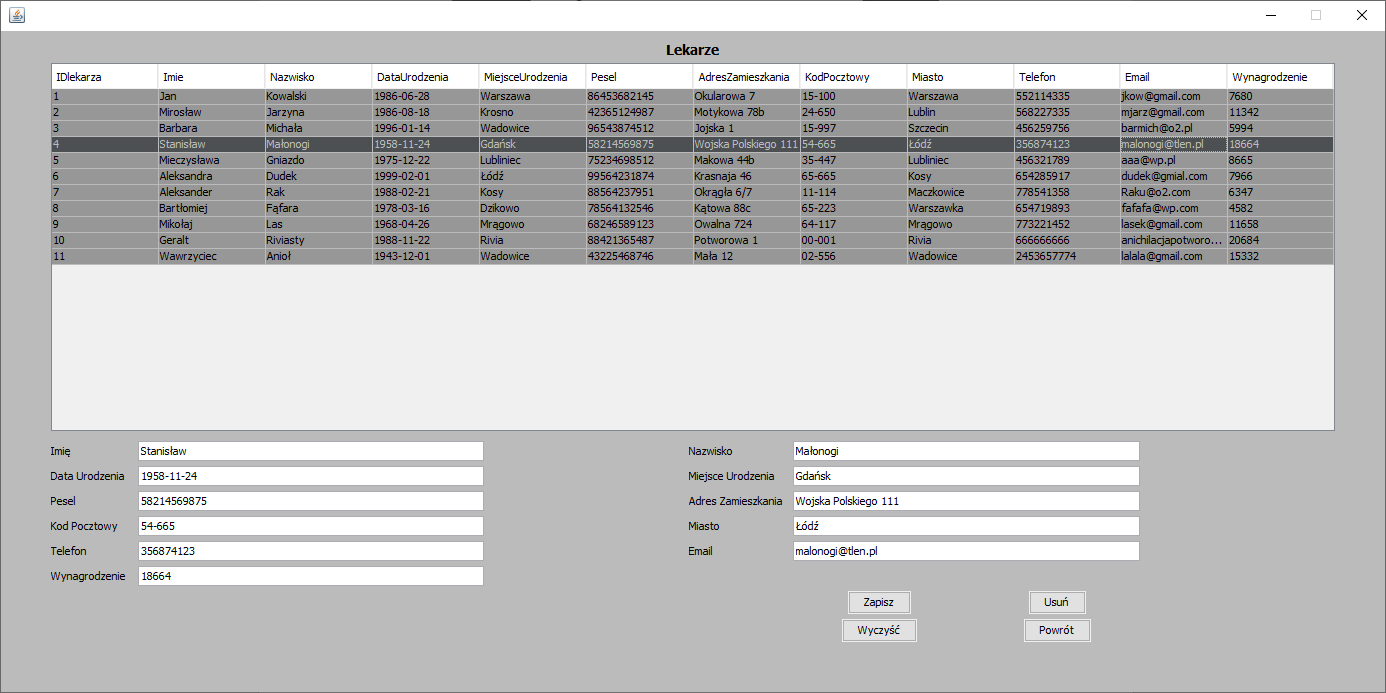


Rysunek 3. Monit wylogowania

Po wybraniu kontekstu „Lekarze” otwiera się panel z bezpośrednią możliwością podglądu   
i edycji danych (Rysunek 4). Wybranie wiersza w tabeli skutkuje pojawieniem się danych   
w oknach formularza co umożliwia edycję danych (Rysunek 5).

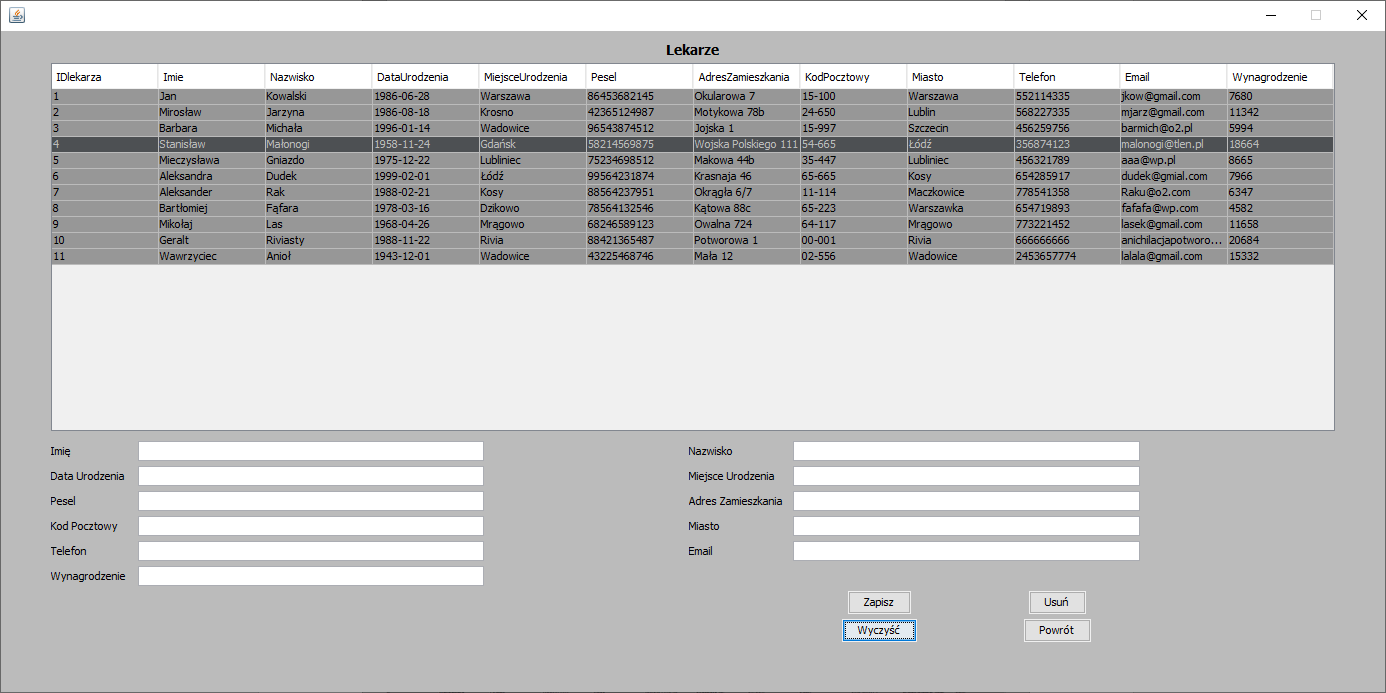


Rysunek 4. Panel „Lekarze”

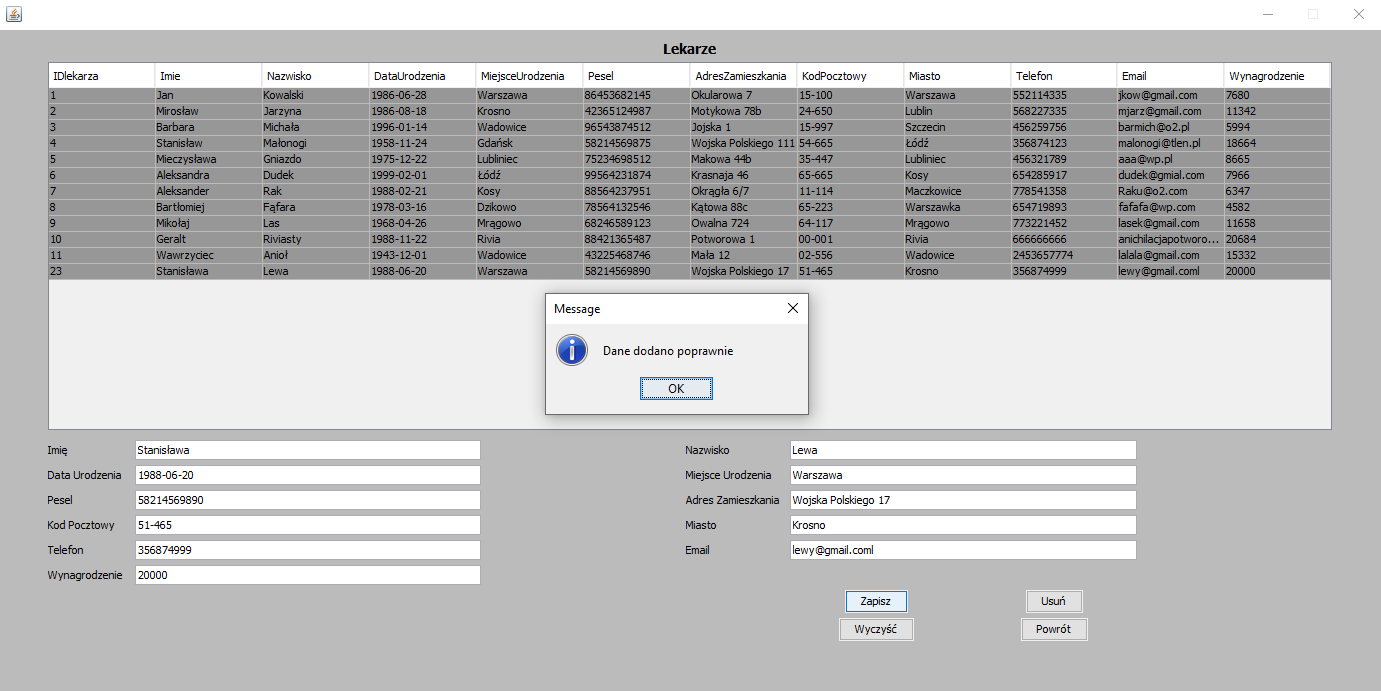


Rysunek 5. Panel „Lekarze”-podgląd danych

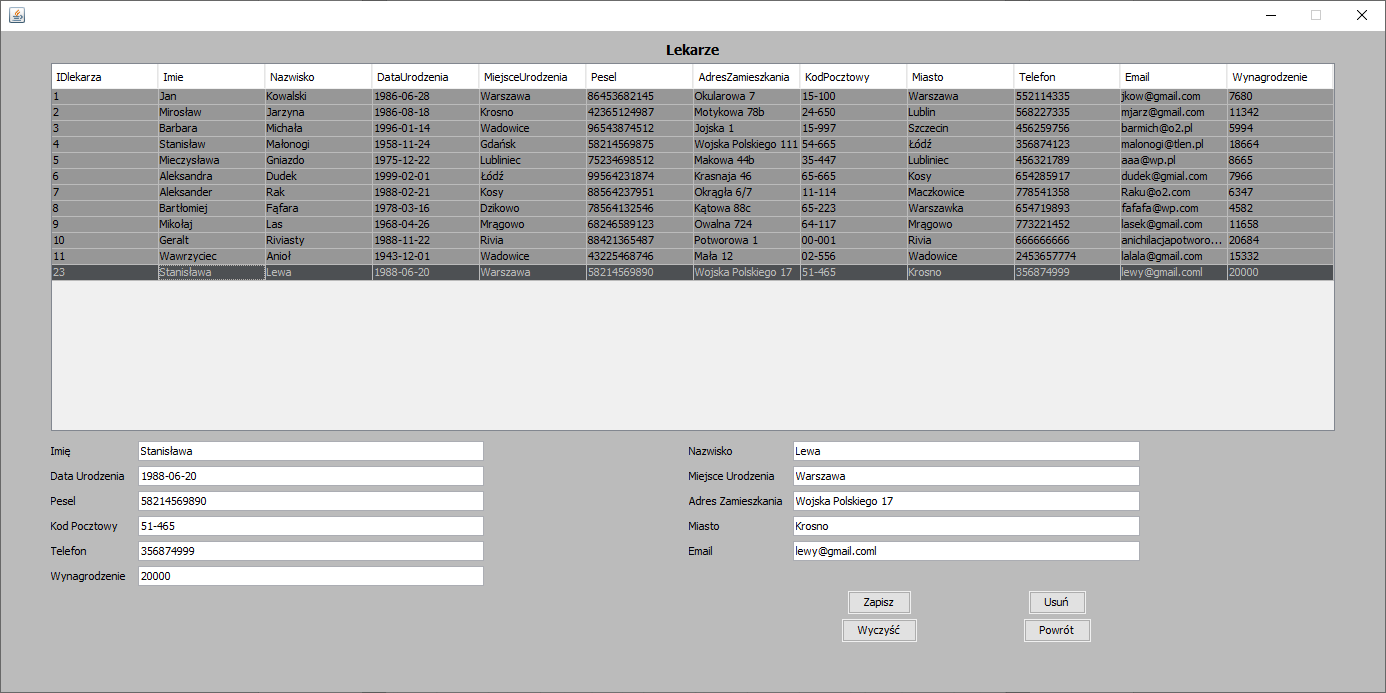
Przycisk „wyczyść” usuwa wszystkie dane z pól formularza (Rysunek 6). Przycisk „zapisz”, po wprowadzeniu poprawnych danych, doda stosowny rekord do bazy danych, wyświetli stosowny komunikat oraz automatycznie odświeży wyświetlaną listę (Rysunek 7, Rysunek 8).



Rysunek 6. Panel „Lekarze”-„wyczyść”

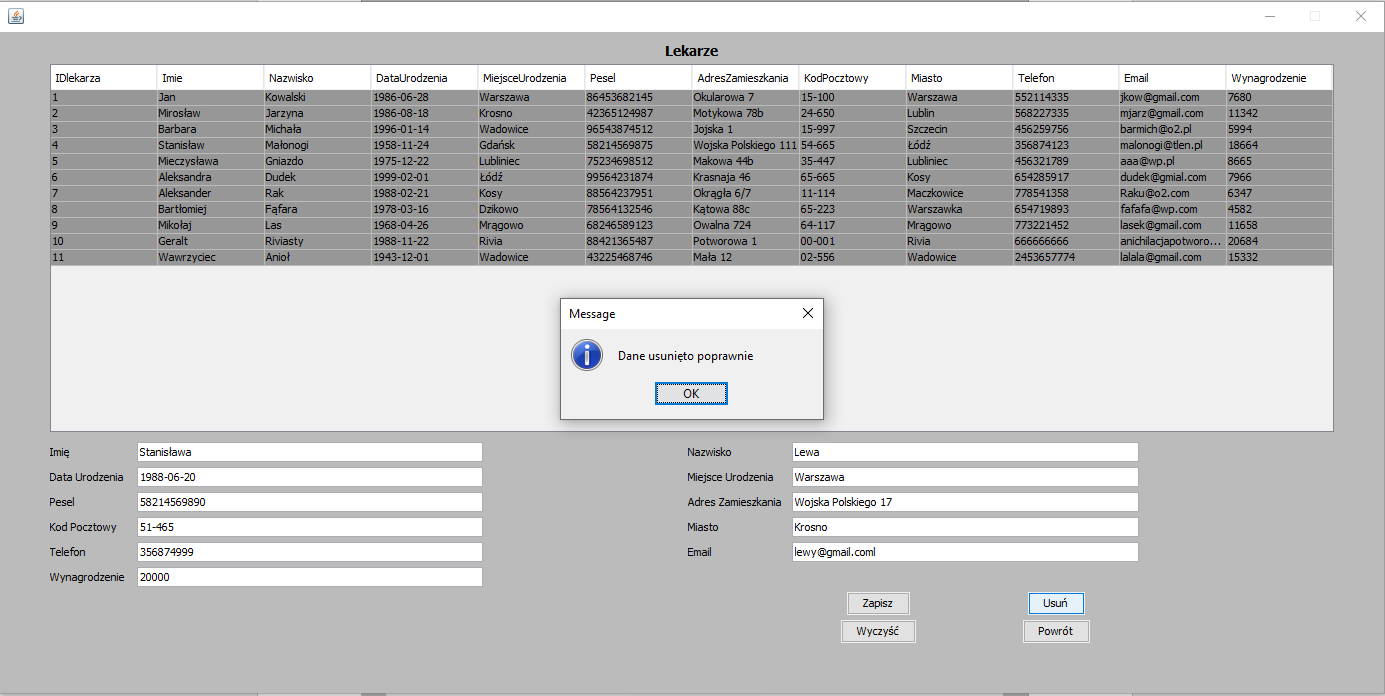


Rysunek 7. Panel „Lekarze”-„zapisz”

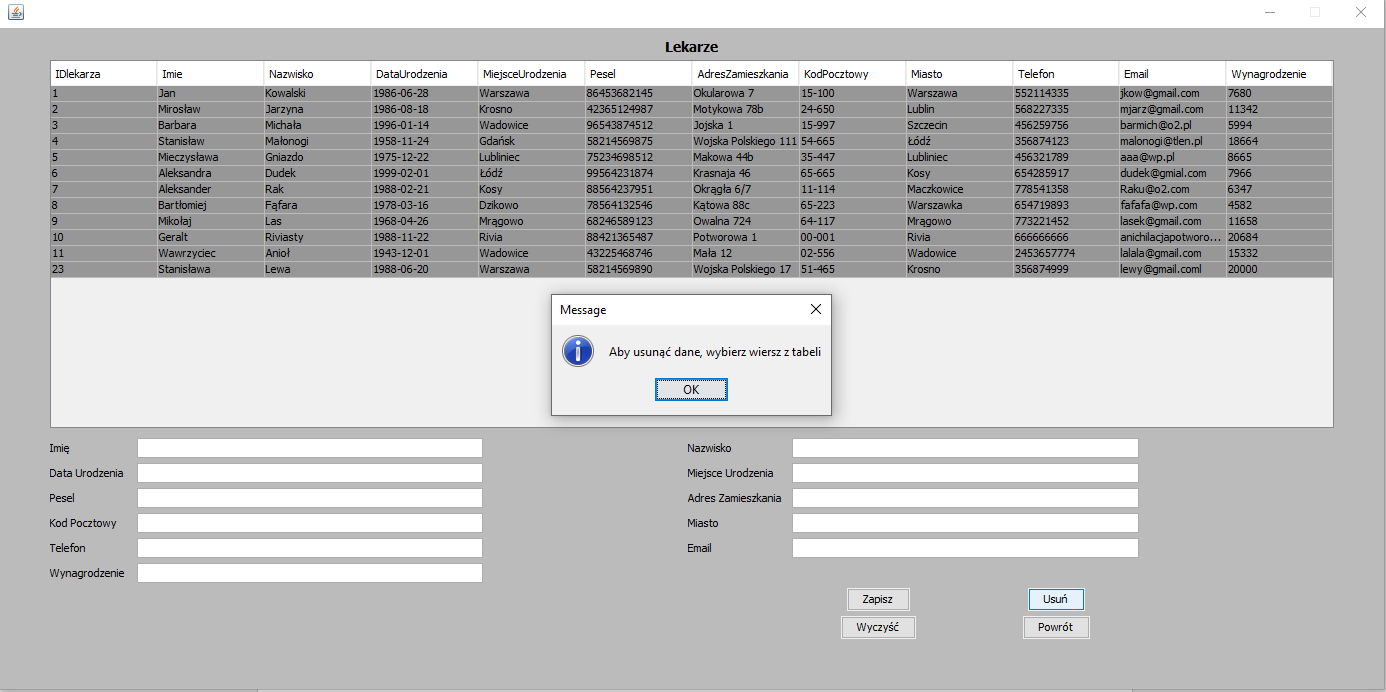


Rysunek 8. Panel „Lekarze”-„zapisz”(cd.)

Przycisk „usuń” usunie rekord z bazy. Użytkownik musi zaznaczyć interesujący go wiersz tabeli (Rysunek 9). W przypadku gdy użytkownik nie zaznaczy wiesza, zostanie  
o to poproszony (Rysunek 10). Przycisk „powrót” wraca użytkownika do głównego menu wyboru.

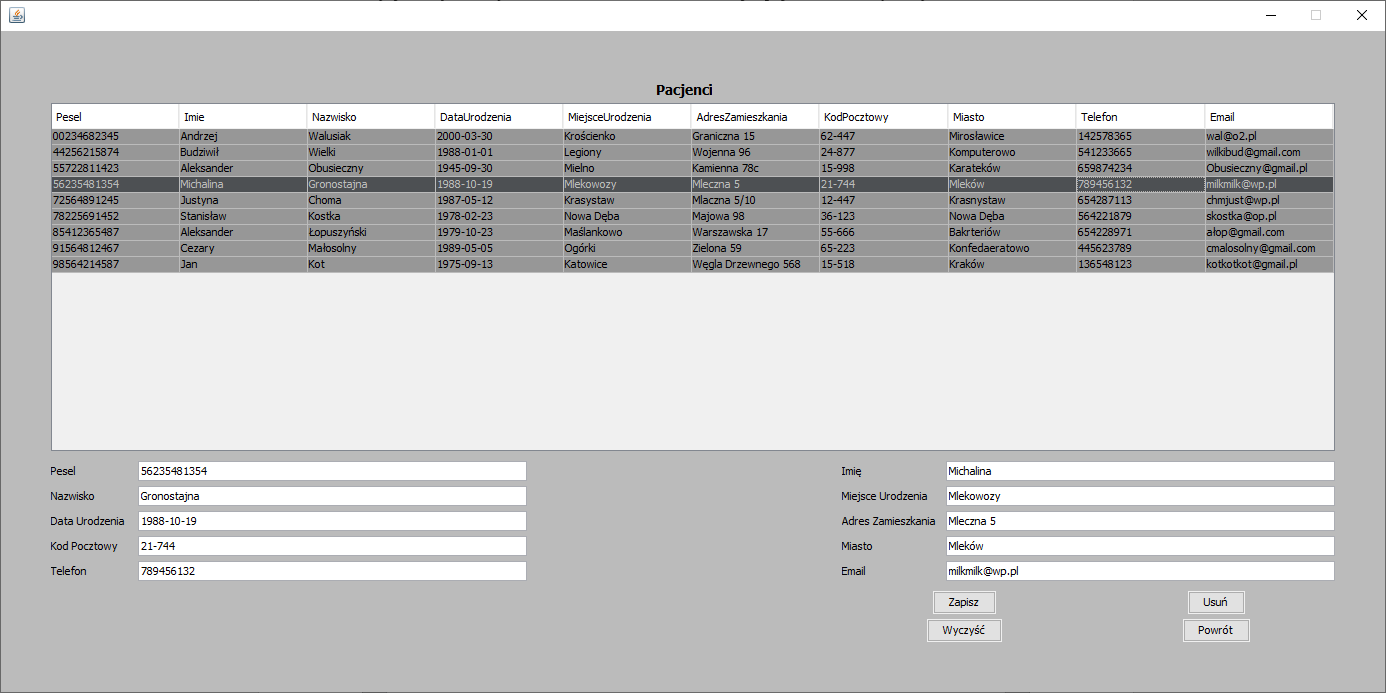


Rysunek 9. Panel „Lekarze”-„usuń”(poprawne usunięcie danych)



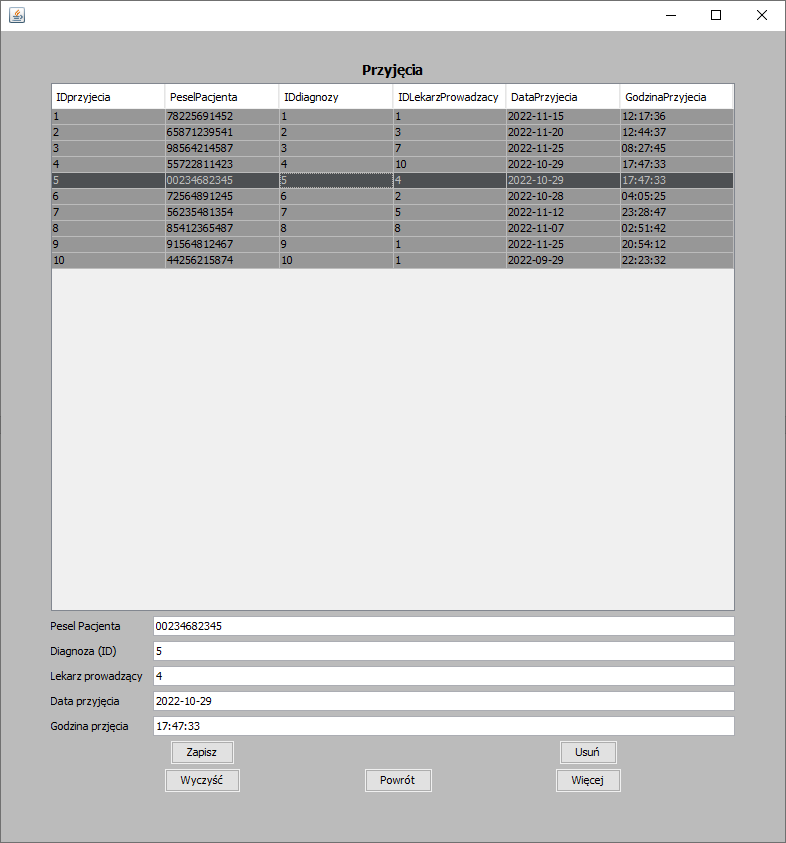
Rysunek 10. Panel „Lekarze”-„usuń”(brak zaznaczonego wiersza)

Po wybraniu kontekstu „Pacjenci” otwiera się panel z bezpośrednią możliwością podglądu  
i edycji danych. Wybranie wiersza w tabeli skutkuje pojawieniem się danych w oknach formularza co umożliwia edycję danych (Rysunek 11). Panele „Lekarze” oraz „Pacjenci” różnią się danymi na których operują oraz wyglądem formularza. Pozostałe funkcjonalności programu (tj. zachowanie przycisków, monity i komunikaty, walidacja danych) są identyczne jak  
 w przypadku panelu „Lekarze”.



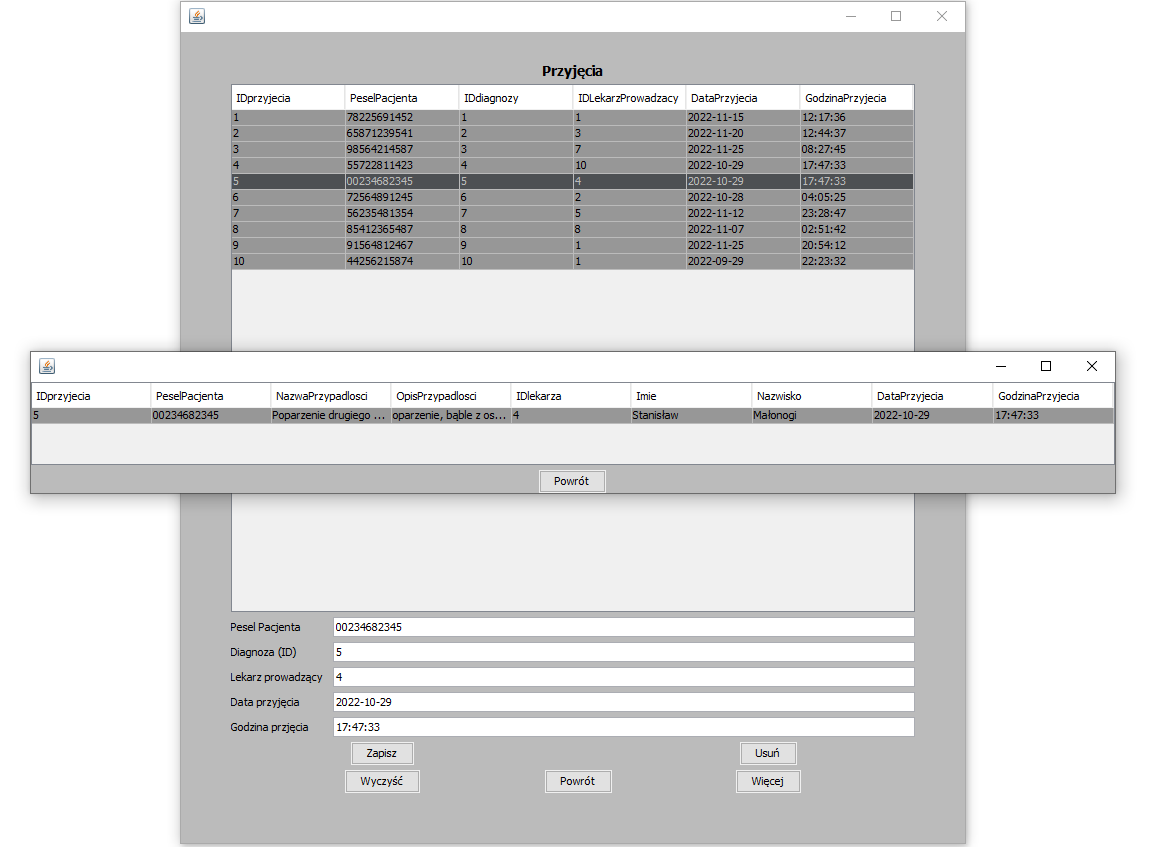
Rysunek 11. Panel „Pacjenci”

Po wybraniu kontekstu „Przyjęcia” z menu głównego otwiera się panel z bezpośrednią możliwością podglądu i edycji danych przyjęć. Wybranie wiersza w tabeli skutkuje pojawieniem się danych w oknach formularza co umożliwia edycję danych (Rysunek 12). Podstawowe funkcjonalności programu (tj. zachowanie przycisków, monity i komunikaty, walidacja danych) są identyczne jak w przypadku poprzednich paneli.



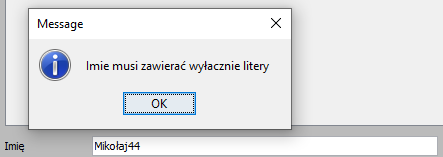
Rysunek 12. Panel „Przyjęcia”

Sprawą indywidualną dla tego panelu jest przycisk „więcej” który umożliwia podgląd wskazanego przez użytkownika wiersza tabeli w powiększeniu. Ukazujące się okno wyświetla szczegółowe dane konkretnego przypadku łącznie z dokładnym typem i opisem przypadłości, podstawowymi danymi lekarza prowadzącego i pacjenta (Rysunek 13).

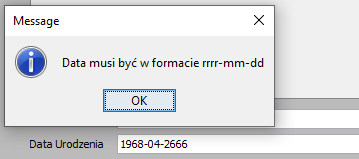


Rysunek 13. Panel „Przyjęcia”-„więcej”

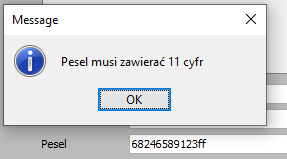
Wszystkie pola formularzy, z każdego z wymienionych paneli posiadają walidację danych w celu uniemożliwienia wprowadzenia niepoprawnych danych do bazy. Rysunki 14,15,16,17 przedstawiają przykładowe komunikaty ukazujące się użytkownikowi po próbie wprowadzenia niepoprawnych danych.



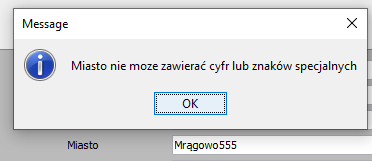
Rysunek 14. Walidacja danych formularza-pole Imię



Rysunek 15. Walidacja danych formularza-pole Data Urodzenia



Rysunek 16. Walidacja danych formularza-pole Pesel

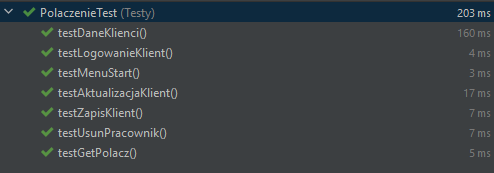


Rysunek 17. Walidacja danych formularza-pole Miasto

Są to komunikaty przykładowe, każde pole formularza posiada osobny-monit dostosowany do wymaganych danych.

1. **Testy jednostkowe**

Zostały wykonane testy jednostkowe dla każdej klasy tak aby sprawdzić działanie pojedynczych funkcji. Test testGetPolacz sprawdza czy istnieje takie połączenie jakie zadeklarowaliśmy w funkcji. Dla metody logowanie, zapis, aktualizacja, usuń zostały przeprowadzone testy pozwalające na podanie przykładowych danych aby sprawdzić czy dane te są zapisywane, aktualizowane, usuwane oraz czy logowanie przynosi oczekiwany rezultat są to testy o nazwach (testDaneKlienci, testLogowanieKlient, testAktualziacjaKlient, testZapisKlient, testUsunPracownik). Ostatni test testMenuStart wybiera przykładową wartość oraz sprawdza czy zostaje wywołana odpowiednia metoda.



Rysunek 16. Testy

1. **Repozytorium, system kontroli wersji**

Projekt został zrealizowany z wykorzystaniem systemu kontroli wersji Git. Dokumentacja oraz projekt został umieszczony w repozytorium dostępnym pod adresem:

<https://github.com/citizen555/szpital-Java-Project>

1. **Podsumowanie**

Dokumentacja ma na celu przedstawienie w sposób zrozumiały dla odbiorcy działanie projektu oraz jego stronę techniczna. Aplikacja spełnia wszystkie założenia oraz jest łatwa do obsługi, rozbudowy lub integracji z innym rozwiązaniem.

1. **Literatura**

1. <https://stackoverflow.com> (data dostępu: 10.01.2023)
2. <https://docs.oracle.com/javase/tutorial/index.html> (data dostępu: 05.01.2023)
3. <https://github.com> (data dostępu 10.01.2023)
4. https://www.youtube.com/ (data dostępu 03.01.2023)

1. <https://chat.openai.com/> (data dostępu 22.01.2023)