

Warszawa, 22.01.2020

Politechnika Warszawska
Wydział Elektroniki i Technik Informacyjnych

Informatyczne systemy medyczne (ISMED) - projekt
Użycie bazy danych do przechowywania informacji medycznych

Dokumentacja końcowa

Prowadzący: dr inż. Tymon Rubel
Wykonawcy: Adamiuk Zuzanna, Krakowiak Aleksandra

System obsługi bazy badań krwi

Autorzy: Adamiuk Zuzanna, Krakowiak Aleksandra

Treść polecenia:

System rejestracji wyników badań laboratoryjnych. Zadaniem aplikacji jest zbieranie danych pacjentów oraz wartości pięciu parametrów oznaczanych w badaniu krwi (np. liczby białych i czerwonych, OB, CRP i glukoza). Dla wybranego pacjenta program musi udostępniać listę wszystkich wykonanych badań, natomiast dla wybranego badania – generować czytelny raport zawierający wyniki wszystkich oznaczonych parametrów.

Środowisko: IntelliJ IDEA

Język programowania: Java 11

Przechowywanie plików bazy: Apache Derby

Sposób działania:

Aplikacja umożliwia dodawanie rekordów do bazy danych przechowującej następujące informacje o pacjentach:

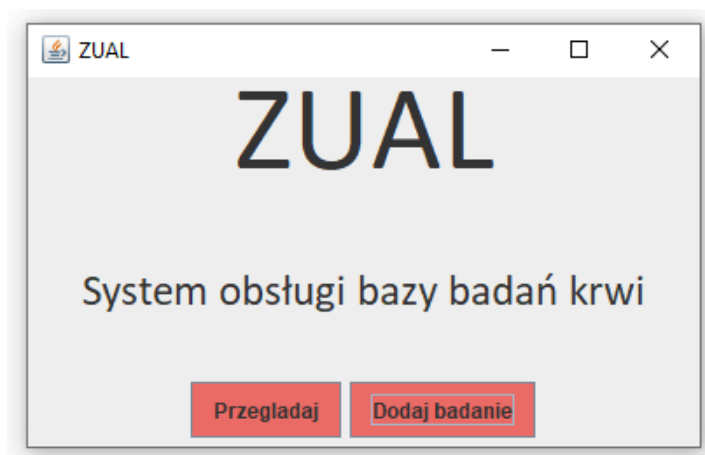
- PESEL
- nazwisko
- imię
- płeć
- wiek

oraz do bazy badań przechowującej następujące informacje:

- nr ID badania
- PESEL pacjenta
- data wykonania badania
- leukocyty (WBC) [tys/ μ l]
- erytrocyty (RBC) [mln/ μ l]
- trombocyty (PLT) [tys/ μ l]
- monocyty (MONO) [tys/ μ l]
- limfocyty (LYM) [tys/ μ l].

Opis działania aplikacji:

Wyniku kompilacji aplikacji, tworzy się główny panel aplikacji (Rys.1) oraz tworzona jest baza danych z przykładowymi rekordami (jeśli takowa jeszcze nie istnieje). Użytkownik po utworzeniu się panelu głównego, może zdecydować, czy chce przeglądać wyniki badań danego pacjenta (znając jego numer PESEL) lub dodać nowe badanie i/lub pacjenta (jeśli takowy nie znajduje się w bazie).

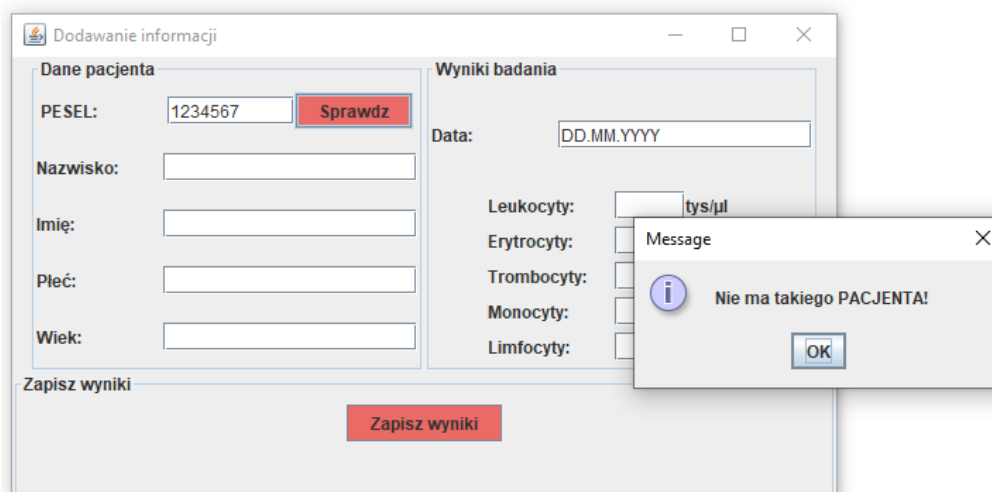


Rys. 1. Widok panelu głównego aplikacji

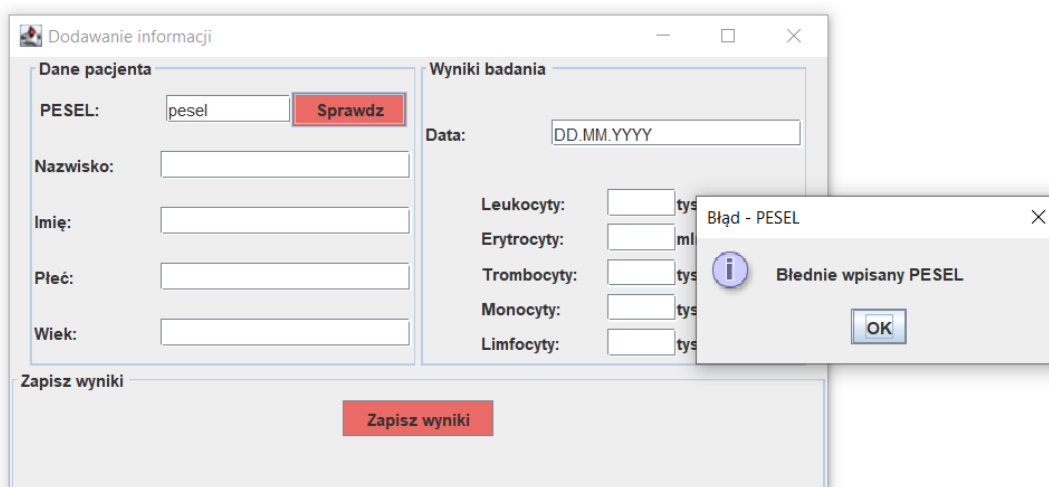
Jeśli użytkownik wybierze opcję „Dodaj badanie” zostanie wyświetlony kolejny panel z możliwością dodawania kolejnych rekordów. (Rys.2). Panel zawiera w sobie przykładowe dane, aby użytkownik wiedział w jaki sposób wprowadzać dane.

Rys. 2. Widok panelu z możliwością dodawania kolejnych rekordów

Panel składa się z 3 części. Część *Dane Pacjenta* odpowiada za wprowadzenie informacji o pacjencie. W tym podpanelu znajduje się przycisk *Sprawdź*, jego zadaniem, jest sprawdzanie czy dany PESEL (w bazie danych Pacjenci rozróżniani są ze względu na PESEL) istnieje w bazie. Na Rys. 3 i Rys. 4 pokazano co się dzieje w przypadku, gdy dany PESEL nie znajduje się w bazie oraz gdy dany PESEL został niepoprawnie wpisany.



Rys. 3. Widok panelu i komunikatu, jeśli sprawdzany PESEL nie znajduje się w bazie



Rys. 4. Widok panelu i komunikatu, jeśli sprawdzany PESEL jest błędnego formatu

Jak widać na powyższych rysunkach, gdy wpisany PESEL nie znajduje się w bazie lub jego format jest błędny, użytkownik zostaje o tym poinformowany. W przypadku, gdy szukany PESEL znajduje się w bazie danych to dane o pacjencie, uzupełniają się automatycznie (Rys. 5). Jeśli dany PESEL nie znajduje się w bazie, użytkownik uzupełnia pola tekstowe ręcznie.

The screenshot shows a window titled "Dodawanie informacji" (Adding information). It is divided into two main sections: "Dane pacjenta" (Patient data) and "Wyniki badania" (Research results).

Dane pacjenta:

- PESEL: 980516 (with a red "Sprawdz" button next to it)
- Nazwisko: Krakowiak
- Imię: Aleksandra
- Płeć: K
- Wiek: 22

Wyniki badania:

- Data: DD.MM.YYYY
- Leukocyty: [] tys/μl
- Erytrocyty: [] mln/μl
- Trombocyty: [] tys/μl
- Monocyty: [] tys/μl
- Limfocyty: [] tys/μl

At the bottom, there is a red button labeled "Zapisz wyniki" (Save results).

Rys. 5. Widok panelu, jeśli PESEL znajduje się w bazie

Kolejną część omawianego panelu to *Wyniki badania*, ten obszar odpowiada za pobranie danych od użytkownika odnośnie wyników przeprowadzonego badania krwi. Należy podać datę wykonania badania oraz wartości badanych parametrów. Dodatkowo, program informuje o tym, czy typ wprowadzonych danych jest poprawny – komunikat wyświetla się, gdy dane są niepoprawne (Rys. 6) lub pole tekstowe nie zostało uzupełnione. Jeśli komunikat wystąpił, pola tekstowe resetują się (Rys. 2).

This screenshot shows the same "Dodawanie informacji" window as in Figure 5, but with a "Message" dialog box overlaid on the right. The dialog box has an information icon and the text "Błędne dane Badania" (Incorrect research data), with an "OK" button at the bottom.

In the background window, the "Wyniki badania" section now contains numerical values:

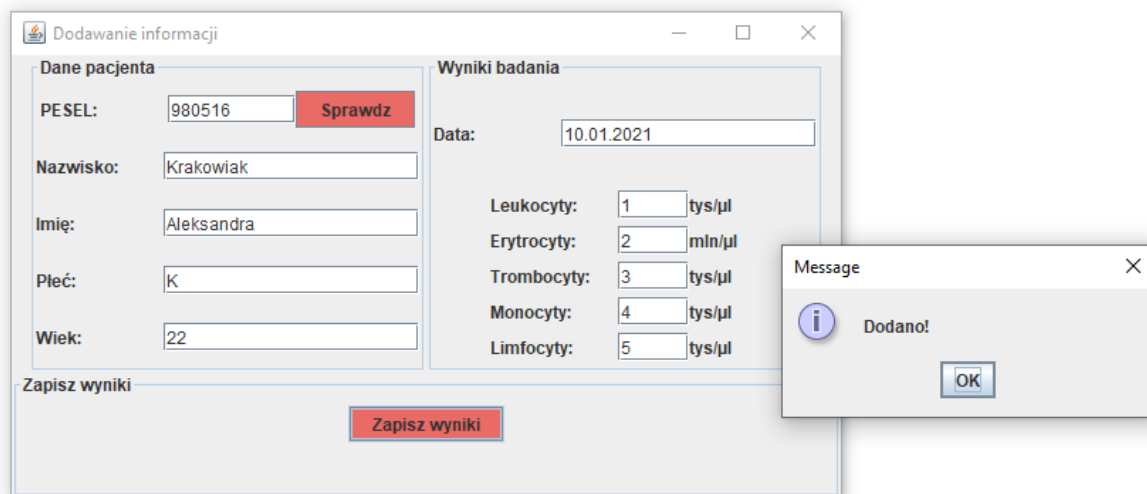
- Data: 10.01.2021
- Leukocyty: 1 tys/μl
- Erytrocyty: 2 mln/μl
- Trombocyty: c tys/μl
- Monocyty: 3 tys/μl
- Limfocyty: 4 tys/μl

The "Zapisz wyniki" button remains at the bottom.

Rys. 6. Widok panelu i komunikaty, jeśli dane badania zostały wprowadzone nieprawidłowo

Ostatnią częścią tego panelu jest *Zapisz wyniki*, w tym obszarze znajduje się przycisk *Zapisz wyniki*. Przycisk ten odpowiada za wprowadzenie pobranych danych do bazy danych, to on decyduje czy pacjent zostanie wprowadzony do bazy danych (jeśli pacjent istnieje to jego informacje nie zostają ponownie wysyłane do bazy). Jeśli wszystkie dane zostały

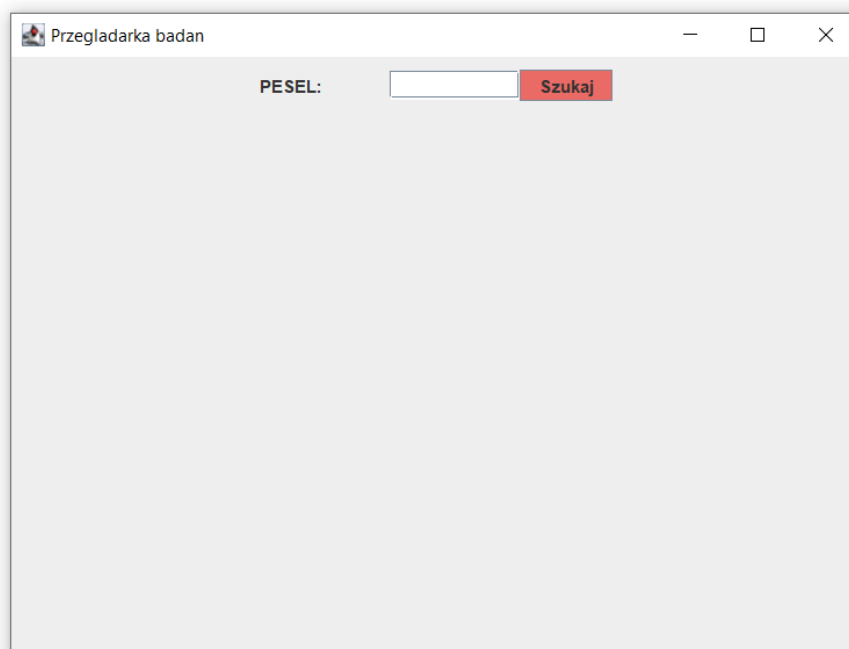
wprowadzone poprawnie to pojawia się komunikat przedstawiony na Rys. 7. Po dodaniu badania i/lub pacjenta, pola tekstowe resetują się. (Rys. 2)



Rys. 7. Widok panelu i komunikatu o poprawnie dodanych danych do bazy danych

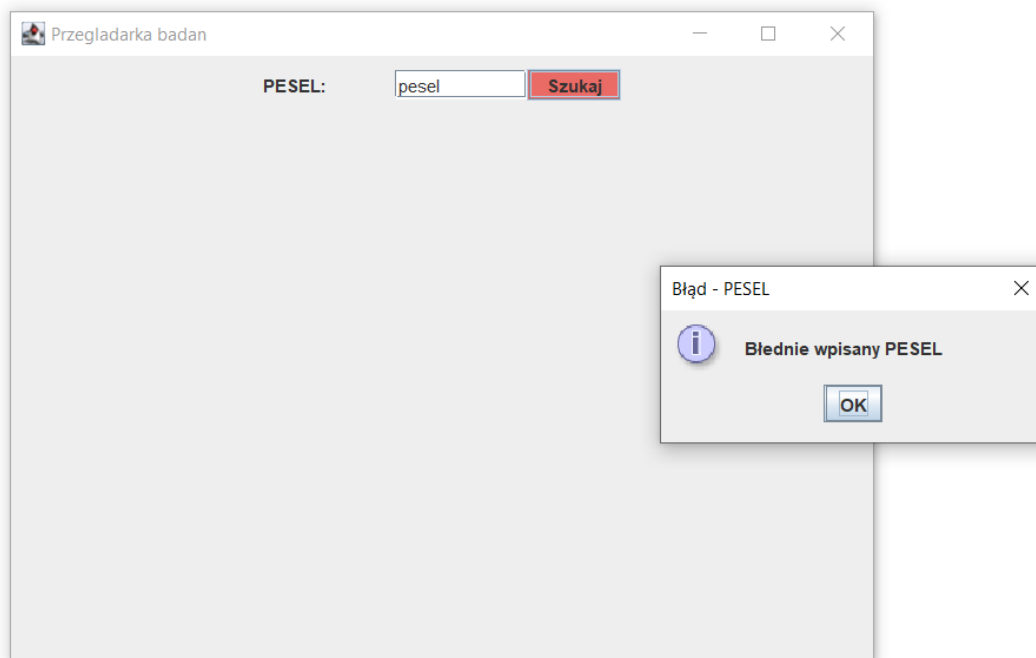
Gdy użytkownik, będzie chciał zakończyć pracę z tym panelem, powinien zamknąć panel przy pomocy „X”.

Jeżeli jednak użytkownik zamiast opcji “Dodaj badanie” wybierze opcję “Przeglądaj” z panelu startowego (Rys. 1), zostanie wyświetlone nowe okno (Rys. 8). W celu obejrzenia wyników badań, użytkownik musi wprowadzić PESEL pacjenta, którego badania chce zobaczyć.

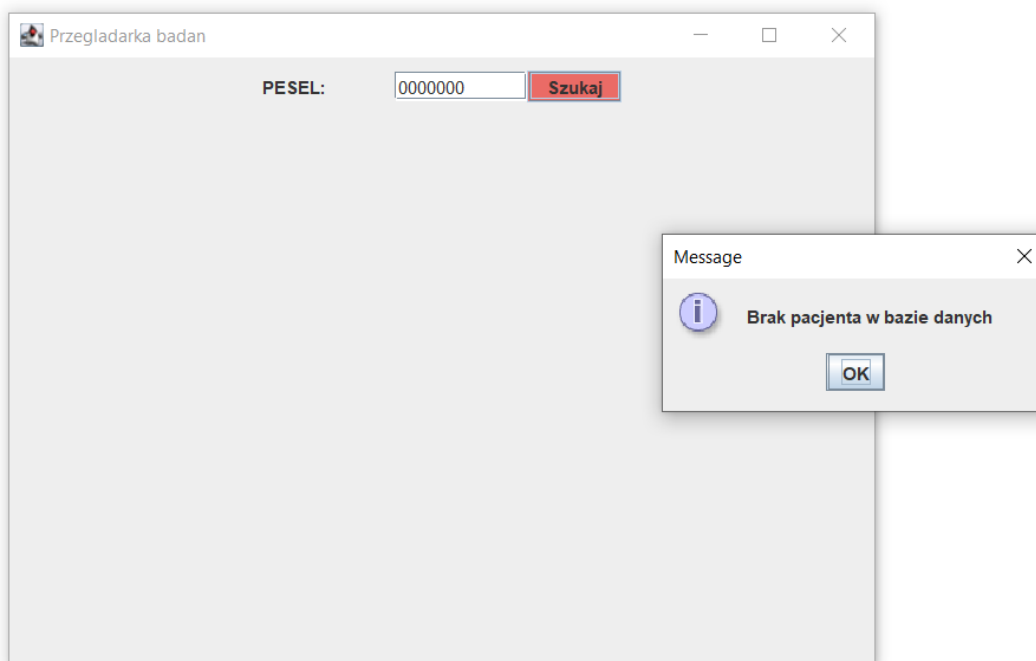


Rys. 8. Widok panelu przed wprowadzeniem numeru PESEL

Wprowadzanie numeru PESEL zostało wyposażone w obsługę błędów, o których użytkownik jest informowany poprzez okna dialogowe. Informują one zarówno o błędnym formacie wprowadzonych danych (Rys. 9), jak i o braku pacjenta o podanym numerze PESEL w bazie (Rys. 10).



Rys. 9. Komunikat wyświetlany przy błędnym wprowadzeniu numeru PESEL

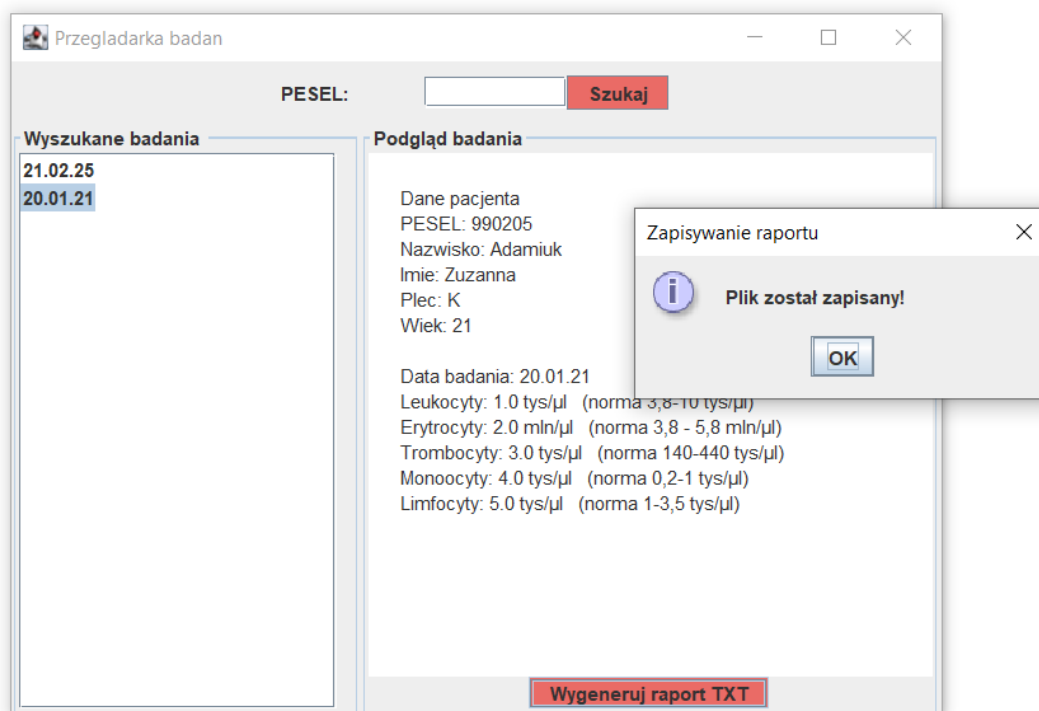


Rys. 10. Komunikat wyświetlany przy braku numeru PESEL w bazie

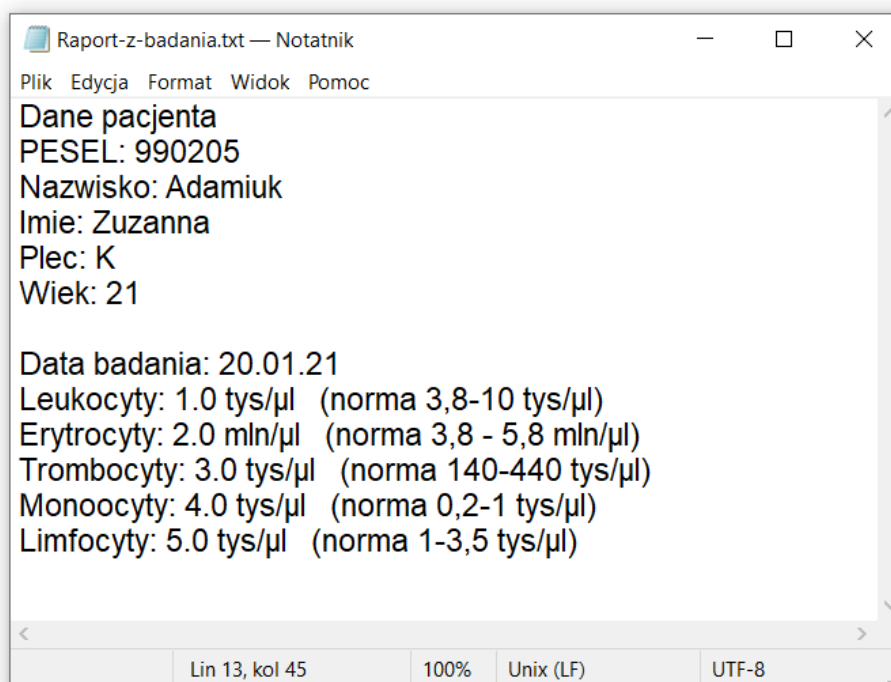
Po wprowadzeniu poprawnego numeru PESEL w lewym panelu wyświetla się lista badań danego pacjenta (wyświetlają się daty poszczególnych badań), a w prawym – podgląd badania oraz przycisk „Wygeneruj raport TXT”, który umożliwia wyeksportowanie pola tekstowego z podglądem wybranego badania do pliku o rozszerzeniu .txt (Rys. 11).

Rys. 11. Widok panelu po wprowadzeniu numeru PESEL i wybraniu badania

Po wybraniu opcji „Wygeneruj raport TXT”, zostaje wyświetlony komunikat potwierdzający wybranie danej opcji (Rys. 12), a tekst zostaje zapisany do pliku o nazwie Raport-z-badania.txt (Rys. 13) mieszczącym się w katalogu z kodem źródłowym aplikacji.



Rys. 12. Komunikat wyświetlany po wybraniu opcji „Wygeneruj raport TXT”.



Rys. 13. Raport z wybranego badania – plik Raport-z-badania.txt

Ogólna struktura kodu:

Klasy:

- Model:
metody umożliwiające połączenie z bazą danych, a następnie za dodawanie i sprawdzanie rekordów w bazie
- Pacjent:
metody *get* oraz *set* dla cech (PESEL, imię, nazwisko, płeć, wiek)
- Badanie:
metody *get* oraz *set* dla cech (PESEL, ID badania, data wykonania, leukocyty, erytrocyty, trombocyty, monocyty, limfocyty)
- WidokGui:
Biblioteka Swing
klasa odpowiedzialna za wygląd interfejsu
- ZdarzenieGui:
klasa odpowiedzialna za wykonywanie zdarzeń: dodawanie informacji o pacjencie oraz wartości parametrów w badaniu krwi, wyszukiwanie pacjenta i wyświetlanie jakie badania są do niego przepisane oraz generacja wyników wybranego badania (jako plik TXT)
- Main:
klasa odpowiedzialna za poprawne uruchamianie aplikacji

Dodatkowe biblioteki:

- biblioteka Swing
- JDBC (Java DataBase Connectivity)
- FileWriter (część biblioteki Java.io)
- Apache Derby (db-derby-10.15.2.0)

Podział zadań:

Aleksandra:

- Stworzenie baz danych - przechowującej dane o pacjentach i drugiej przechowującej parametry badania
- Odpowiedzialność za klasy zdarzenia i widok GUI – przy dodawaniu nowego badania
- Utworzenie klas Pacjent, Badanie i Main
- Poprawność nazewnictwa klas i parametrów

Zuzanna:

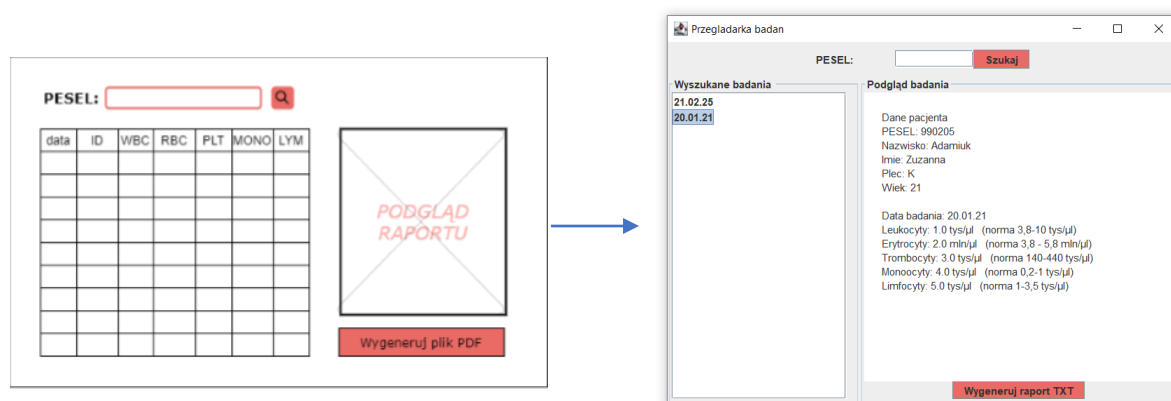
- Odpowiedzialność za klasy zdarzenia i widok GUI – przy generowaniu raportu i podglądu wybranego badania
- Utworzenie szablonu raportu (podgląd i plik TXT)
- Utworzenie strony startowej
- Poprawność nazewnictwa klas i parametrów

Repozytorium GIT

- miejsce, gdzie będą przechowywane pliki wykorzystywane w projekcie połączone z wybranym środowiskiem
- miejsce, gwarantujące pracę na jednakowej wersji aplikacji dla wszystkich użytkowników
- miejsce oddania projektu - umożliwiające prowadzącemu wgląd w nasze postępy pracy

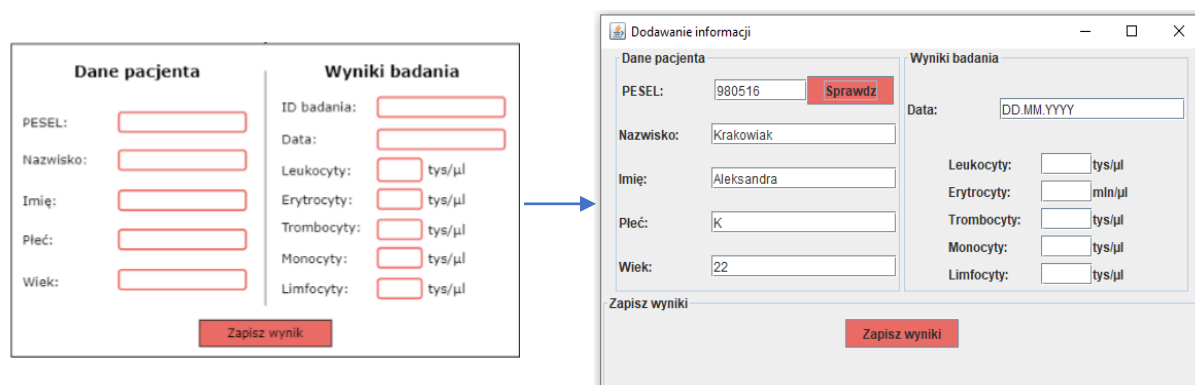
Zmiany względem pierwotnych założeń

Podczas tworzenia kodu źródłowego programu zmieniono kilka założeń, które zastąpiono bardziej wydajnymi oraz praktycznymi rozwiązaniami. W oknie przeglądarki badań dodano podgląd w formie interaktywnej listy (JList) z dodanym ListSelectionListnerem, w pierwotnej wersji, założono, że od razu będą wyświetlane wszystkie informacje dot. badań. Zmieniono również format generowanego raportu – pierwotnie miał to być plik PDF, zdecydowano jednak generowanie pliku TXT.



Rys. 14. Porównanie założeń wyglądu panelu przeglądarki badań z wersją finalną

W oknie dodawania nowego badania dodano automatyczne uzupełnianie pól tekstowych, po wpisaniu numeru PESEL – numer PESEL znajduje się w bazie. W realizacji projektu po kliknięciu przycisku informacje uzupełniają się automatycznie, jeśli PESEL już jest w bazie. Zmiana wyglądu panelu do wprowadzania danych. Został uwzględniony dodatkowy przycisk oraz usunięcie numeru id wprowadzanego badania.



Rys. 15. Porównanie założeń wyglądu panelu dodawania wyników badań z wersją finalną